

UNIVERSIDAD COMPLUTENSE DE MADRID
FACULTAD DE CIENCIAS QUÍMICAS
Departamento de Ingeniería Química y de Materiales



TESIS DOCTORAL

**Aplicación de métodos computacionales a la monitorización y
el cosechado de cultivos de microalgas**

MEMORIA PARA OPTAR AL GRADO DE DOCTOR

PRESENTADA POR

Patricio Roberto López Expósito

Director

Carlos Negro Álvarez

Madrid

Ed. electrónica 2019



UNIVERSIDAD COMPLUTENSE MADRID

FACULTAD DE CIENCIAS QUÍMICAS
DEPARTAMENTO DE INGENIERÍA QUÍMICA Y DE MATERIALES

TESIS DOCTORAL

Aplicación de métodos computacionales a la monitorización y el cosechado de cultivos de microalgas

MEMORIA PARA OPTAR AL GRADO DE DOCTOR EN INGENIERÍA QUÍMICA
PRESENTADA POR

Patricio Roberto López Expósito

DIRECTOR:

Prof. Carlos Manuel Negro Álvarez

MADRID, 2019



U N I V E R S I D A D
COMPLUTENSE
M A D R I D

DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD DE LA TESIS PRESENTADA PARA OBTENER EL TÍTULO DE DOCTOR

D./Dña. Patricio Roberto López Expósito,
estudiante en el Programa de Doctorado de Ingeniería Química,
de la Facultad de Ciencias Químicas de la Universidad Complutense de
Madrid, como autor/a de la tesis presentada para la obtención del título de Doctor y
titulada:

Aplicación de métodos computacionales a la monitorización y el cosechado de cultivos de microalgas

y dirigida por: Carlos Negro Álvarez

DECLARO QUE:

La tesis es una obra original que no infringe los derechos de propiedad intelectual ni los derechos de propiedad industrial u otros, de acuerdo con el ordenamiento jurídico vigente, en particular, la Ley de Propiedad Intelectual (R.D. legislativo 1/1996, de 12 de abril, por el que se aprueba el texto refundido de la Ley de Propiedad Intelectual, modificado por la Ley 2/2019, de 1 de marzo, regularizando, aclarando y armonizando las disposiciones legales vigentes sobre la materia), en particular, las disposiciones referidas al derecho de cita.

Del mismo modo, asumo frente a la Universidad cualquier responsabilidad que pudiera derivarse de la autoría o falta de originalidad del contenido de la tesis presentada de conformidad con el ordenamiento jurídico vigente.

En Madrid, a 16 de mayo de 2019

**LOPEZ
EXPOSITO
PATRICIO
ROBERTO -**
Fdo.: 45580918R

Firmado digitalmente por LOPEZ
EXPOSITO PATRICIO ROBERTO -
45580918R
Nombre de reconocimiento (DN):
c=ES,
serialNumber=IDCES-45580918R,
givenName=PATRICIO ROBERTO,
sn=LOPEZ EXPOSITO, cn=LOPEZ
EXPOSITO PATRICIO ROBERTO -
45580918R
Fecha: 2019.05.16 12:06:37 +02'00'

Esta DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD debe ser insertada en
la primera página de la tesis presentada para la obtención del título de Doctor.

CARLOS MANUEL NEGRO ÁLVAREZ CATEDRÁTICO DEL DEPARTAMENTO DE
INGENIERÍA QUÍMICA Y DE MATERIALES DE LA UNIVERSIDAD COMPLUTENSE
DE MADRID

INFORMA

De que el trabajo de investigación titulado “APLICACIÓN DE MÉTODOS COMPUTACIONALES A LA MONITORIZACIÓN Y EL COSECHADO DE CULTIVOS DE MICROALGAS” ha sido realizado bajo su dirección dentro del Grupo de Investigación de Celulosa y Papel de la Universidad Complutense de Madrid, y constituye la memoria que presenta D. Patricio Roberto López Expósito para optar al Grado de Doctor. A su juicio esta tesis cumple con los requisitos de rigor científico y originalidad para que el alumno pueda optar al título de doctor.

Y para que conste a los efectos oportunos, firma el presente informe, en Madrid a 19 de marzo de 2019.

AGRADECIMIENTOS

A mis padres agradezco todo. Su amor, apoyo y confianza, la educación que me dieron, siempre libre, siempre exenta de intervenciones. Quiero agradecer especialmente a mi madre que me enseñase a tratar y disfrutar el trato con los demás, y a mi padre, que despertase en mí el amor y la curiosidad por la naturaleza. Agradezco a mis hermanos, Paco y Moisés, y a Laura y a Loli su cariño y el aliento que me han prestado para continuar con la investigación doctoral en estos años.

Agradezco a Ángeles y Carlos, su confianza y la oportunidad que me han prestado para demostrar que era digno de ella. Esta tesis, sin embargo, no habría sido posible solo con confianza, puesto que encajar el trabajo de gestión en el grupo con el de investigación supone un esfuerzo ímprobo. Les agradezco pues, especialmente, su consideración y generosidad al ofrecerme espacio y tiempo para poder investigar dispensándome a menudo de tareas, viajes y otros desempeños que ellos han cargado sobre sus espaldas.

A Julio me gustaría agradecer su ingenio y su inquietud por el bienestar de los demás. A Antonio su sentido del humor y sus anécdotas históricas. A Roberto, su teatralidad refrescante. A Jesús, gracias por su disposición a ayudar en todo siempre y por su buen humor y paciencia. A Pablo, gracias por obligarme a programar con objetos. A Emilio, gracias por sus ánimos. A Bárbara, gracias por su ayuda en el laboratorio.

Gracias a Esperanza por enseñarme a cultivar bichos y por su amistad. Gracias a Paloma por su amistad y sus sabios consejos. A Helen, Luis, Rocío e Isa por hacer los días más llevaderos y divertidos. Agradezco a Conchi su amabilidad y al resto de los miembros del grupo su apoyo y paciencia en los momentos duros.

A Antonio, Belén, Ester, Javi, Jhoret, Marcos, Marimar, Miguel, Noemí, Pepelu, Sandra, Silvia y Vicky, gracias por su amistad y los buenos momentos.

A Cristina tengo mucho que agradecerle. Por un lado, que haya comprendido y valorado desde el primer momento el trabajo que he desarrollado en esta tesis y, por otro, que haya compartido conmigo su entusiasmo científico y su buen hacer investigador. Por supuesto, su amistad.

A Olly por tantos y tan entretenidos almuerzos y por su amable insistencia. A Nacho por su amistad y buenos consejos.

Mi agradecimiento al equipo del Centro de Citometría y Microscopía de Fluorescencia (CCMF) de la Universidad Complutense por su buen hacer y por sus consejos sobre procesamiento de imágenes.

Agradezco a las entidades financiadoras de los proyectos CENIT-VIDA (ref. CEN-20101026, Ministerio de Ciencia e Innovación), E4WATER (ref. 280756, Comisión Europea), RETOPROSOST-CM (ref. S2013/MAE-2907, Comunidad de Madrid), por el apoyo económico prestado a las investigaciones asociadas a esta tesis doctoral.

A Paco y Casilda, a mis hermanos.

LISTA DE PUBLICACIONES ORIGINALES

- I. Patricio Lopez-Exposito, Angeles Blanco, Carlos Negro (2016). Estimation of *Chlamydomonas reinhardtii* biomass concentration from chord length distribution data. *J. Appl Phyco*, 28(4), 2315-2322. (JCR, Marine and Fresh Water Biology, 24 de 106, Q1)
- II. Patricio López Expósito, Angeles Blanco, Carlos Negro. (2017). Laser reflectance measurement for the online monitoring of *Chlorella sorokiniana* biomass concentration. *Journal of Biotechnology*, 243, 10-15. (JCR, Biotechnology & Applied Microbiology, 65 de 160, Q2)
- III. López Expósito, P., Campano, C., Blanco, A., & Negro, C. (2019). Microalgae harvesting by the novel flocculant hairy cationic nanocrystalline cellulose. *Colloids and Surfaces B: Biointerfaces*, 178, 329-336. (JCR, Biophysics, 13 de 72, Q1)
- IV. López Expósito, P., Blanco, A., & Negro, C. (2017). Estimating fractal dimension of microalgal flocs through confocal laser scanning microscopy and computer modelling. *Algal Research*, 28, 74-79. (JCR, Biotechnology & Applied Microbiology, 36 de 160, Q1)
- V. Lopez-Exposito, P., Blanco, A., & Negro, C. (2019). Direct estimation of microalgal flocs fractal dimension through laser reflectance and machine learning. *Algal Research*, 37, 240-247. (JCR, Biotechnology & Applied Microbiology, 36 de 160, Q1)

ÍNDICE

RESUMEN EXTENDIDO.....	1
EXTENDED SUMMARY	5
1. INTRODUCCIÓN.....	9
1.1. Las microalgas.....	9
1.1.1 Metabolitos de microalgas con potencial comercial	11
1.1.2. Métodos de cultivo de microalgas	13
1.1.3. Cosechado de microalgas	14
1.1.4. Producción comercial de biomasa de microalgas.....	17
1.2. Floculación de suspensiones de microalgas	18
1.3. Morfología de los agregados de microalgas.....	20
1.3.1. Los objetos fractales y los agregados de partículas	20
1.3.2. La morfología de los agregados de partículas en la separación sólido líquido	22
1.3.3. Caracterización de las suspensiones de sólidos.....	23
2. OBJETIVOS	33
3. MATERIALES Y MÉTODOS EXPERIMENTALES	35
3.1. Cepas de microalgas y su cultivo.	35
3.2. Floculantes.....	37
3.2.1. Quitosano.....	37
3.2.2. Celulosa nanocrystalina hairy cationizada (CNC-HC)	37
3.3. Determinación de la biomasa seca.	37
3.4. Ensayos de floculación	38
3.5. Obtención de datos de distribuciones de tamaños cuerda de las suspensiones	38
3.6. Adquisición de imágenes por microscopía óptica convencional.	40
3.7. Adquisición de imágenes por microscopía confocal láser de barrido	40
4. MÉTODOS COMPUTACIONALES	41
4.1. Modelos de aprendizaje automático aplicados a la regresión.	41
4.2. Algoritmo de optimización de la arquitectura de los modelos de regresión	43
4.2.1. Algoritmo de recocido simulado	43
4.2.2. Generación aleatoria de modelos y su mutación	44

4.2.3. Aplicación del algoritmo de recocido simulado sencillo en la optimización de la arquitectura del perceptrón multicapa	45
4.2.4. Combinación del algoritmo de recocido simulado con validación cruzada	47
4.3. Generación de agregados fractales virtuales.	48
4.3.1. Algoritmo para la combinación de pares de agregados	49
4.3.2. Algoritmo de crecimiento de agregados	50
4.4. Adquisición de datos de los agregados fractales virtuales	52
4.4.1. Algoritmo de sedimentación de los agregados virtuales	52
4.4.2. Estimación de la dimensión fractal en 2D de las proyecciones de agregados virtuales	52
4.4.3. Generación de espectros de tamaños de cuerda de los agregados virtuales	53
4.5. Métodos para la estimación de la dimensión fractal media de suspensiones reales de flóculos de microalgas	54
4.5.1. Método para la estimación de la dimensión fractal media de agregados de microalgas a partir del procesamiento de imágenes de microscopía láser confocal y modelado por ordenador	54
4.5.2. Metodología para el desarrollo del sistema de estimación de la dimensión fractal media de los agregados a través de un modelo de regresión por aprendizaje automático.	56
5. RESULTADOS	59
5.1. Métodos para la estimación de la concentración de la biomasa de microalgas.	62
5.1.1. Estimación de la concentración de <i>Chlamydomonas reinhardtii</i> por medio de redes neuronales artificiales.	62
5.1.2. Estimación de la concentración de la biomasa de <i>Chlorella sorokiniana</i> floculada por medio de bosques aleatorios y máquinas de vectores de soporte	63
5.2. Influencia de geometría de los agregados de microalgas en la filtrabilidad de las suspensiones de biomasa floculada.....	65
5.2.1. Floculación de suspensiones de <i>Chlorella sorokiniana</i> con CNC-HC	65
5.2.2. Efecto de la dosis de CNC-HC sobre el potencial Z de la muestra.	66
5.2.3 Efecto de la dosis de CNC-HC sobre la morfología de los agregados	67
5.2.4. Efecto de la morfología sobre la filtrabilidad de las suspensiones de microalgas floculadas.	71
5.3. Método de estimación de la dimensión fractal en 3D de flóculos de microalgas proyección mediante microscopía laser confocal y modelado por ordenador.	73
5.3.1. Adquisición de imágenes microscópicas de suspensiones de agregados de microalgas y su caracterización morfológica	73
5.3.2. Generación de flóculos virtuales con D_f y máximo diámetro de Feret consignados	76
5.3.3. Proyección de flóculos virtuales y estimación del área proyectada máxima.	78

5.3.4. Inferencia de la D_f por comparación entre dimensiones fractales proyectadas	79
5.4. Desarrollo de un modelo de estimación de la D_f a partir de datos de tamaño de cuerda de suspensiones de microalgas.....	80
5.4.1. Obtención de tamaños de cuerda de flóculos virtuales	80
5.4.2. Entrenamiento y selección del modelo de estimación de la D_f por medio de la DTC... ..	82
5.4.3. Validación del modelo con datos reales.....	82
6. CONCLUSIONES	85
7. REFERENCIAS	89
ANEXO I: PRINCIPALES BLOQUES DE CÓDIGO IMPLEMENTADOS.....	105
ANEXO II: PUBLICACIONES ORIGINALES	153

ÍNDICE DE TABLAS

Tabla 1.1. Contenido en proteínas, carbohidratos y lípidos de biomasa de algunas cepas de microalgas expresado como porcentaje de biomasa seca	10
Tabla 4.1. Hiperparámetros de la red neuronal de tipo perceptrón.....	42
Tabla 4.2. Hiperparámetros de la regresión por máquina de vectores de soporte	42
Tabla 4.3. Hiperparámetros de la regresión por bosques aleatorios	42
Tabla 5.1. Hiperparámetros del perceptrón multicapa seleccionado.....	63
Tabla 5.2. Hiperparámetros del bosque aleatorio de regresión seleccionado.....	64
Tabla 5.3.- Hiperparámetros del modelo de regresión con máquinas de vectores de soporte.....	65
Tabla 5.4. Dimensiones fractales proyectadas promedio (D_2) obtenidas en las condiciones de agitación consideradas	75
Tabla 5.5. Dimensiones fractales proyectadas promedio (D_2) obtenidas en cada familia de agregados para las dimensiones fractales de consigna consideradas.	80
Tabla 5.6. Hiperparámetros seleccionados para el modelo de bosque aleatorio para la estimación de la dimensión fractal.	82

ÍNDICE DE FIGURAS

Figura 1.1. Estructuras moleculares de quitina y quitosano	19
Figura 1.2. Producción de CNC-HC a partir de celulosa	20
Figura 3.1. a) Incubadora termostatzada con iluminación LED; b) fotobiorreactores de laboratorio empleados	36
Figura 3.2. Dispositivo experimental para el cultivo de microalgas en fotobiorreactores	36
Figura 3.3. Equipos de análisis del tamaño de cuerda, a) Sonda FBRM M500L y b) PT G400.....	38
Figura 3.4. Esquema de funcionamiento de la sonda FBRM	39
Figura 3.5. Microscopio Axio Lab.A1 y portamuestras con pocillo	40
Figura 4.1. Algoritmo de recocido simulado para la optimización de la arquitectura del perceptrón....	46
Figura 4.2. Algoritmo de optimización de la arquitectura de modelos de aprendizaje automático mediante la combinación de un algoritmo de optimización por recocido simulado y una validación cruzada.....	48
Figura 4.3. Algoritmo de crecimiento de agregados virtuales	51
Figura 4.4. Esquema conceptual de la estimación de la D_f media de suspensiones de agregados por microscopía y modelado por ordenador	55
Figura 4.5. Procedimiento seguido para el desarrollo de un modelo de estimación de la D_f media de suspensiones de flóculos con distribuciones de tamaño.	57
Figura 5.1. Resumen gráfico de las técnicas más relevantes empleadas en las publicaciones asociadas a la presente tesis.....	61
Figura 5.2. Concentración de biomasa estimada con el modelo de perceptrón frente a la medida en los datos de validación.....	62
Figura 5.3. Concentración de biomasa estimada con el modelo de bosques aleatorios frente a la medida en los datos de validación	64
Figura 5.4. Concentración de biomasa estimada con el modelo de máquinas de soporte de vectores frente a la medida en los datos de validación.....	65
Figura 5.5. Potencial-Z y reducción de absorbancia en función de la dosis de CNC-HC.	66
Figura 5.6. Imágenes microscópicas de suspensiones de <i>C. sorokiniana</i> floculadas con 27,5 ppm de CNC -HC aplicando velocidades de agitación de a) 100 rpm; b) 200 rpm; y c) 300 rpm. Las imágenes de la izquierda corresponden a la fase de floculación y las de la derecha a la de refloculación.	67
Figura 5.7. D_z en función de la dosis para las velocidades de agitación consideradas.....	68
Figura 5.8. Posible unión mediada por CNC-HC intercelular en función de la concentración de nanopartículas: a) baja, b) media, c) alta concentración y d) estructura de un flóculo de microalga inducida con una dosis de CNCC cerca del PIE.	70
Figura 5.9. Volumen de filtrado frente al tiempo para las tres presiones de vacío consideradas: a) 0,5 bar, b) 0,65 bar y c) 0,8 bar; y d) reducción de absorbancia en función de la presión de vacío.	71
Figura 5.10. Eliminación de la absorbancia a distintas presiones de vacío en función de la velocidad de agitación en la floculación.	72
Figura 5.11. Imágenes de microalgas floculadas a distintas velocidades de agitación obtenidas por microscopía láser confocal.....	73
Figura 5.12. Imágenes binarizadas de agregados de <i>C. sorokiniana</i> floculados a 200, 350 y 500 rpm. 74	

Figura 5.13. Dimensión fractal proyectada en función del tamaño de agregados obtenidos a a) 200, b) 350 y c) 500 rpm.	75
Figura 5.14. Agregados virtuales con MDF 307 μm y D_f de consigna entre 1,2 y 2,1.	76
Figura 5.15. DFM de los agregados virtuales en función del DFM medidos.	77
Figura 5.16. Número de partículas en el agregado en función de su MDF de consigna.	77
Figura 5.17. Cuatro proyecciones aleatorias de un agregado virtual con $D_f=2,1$ y MDF 107.	78
Figura 5.18. Proyecciones de un agregado virtual con $D_f=2,1$ y MDF 107 tras la rotación alrededor de su eje máximo.	78
Figura 5.19. Área de la proyección del floculo con $D_f=2,1$ y MDF 107 μm en función del ángulo de giro.	79
Figura 5.20. D_2 de cada agregado virtual para las familias a) 200, b) 350 y c) 500 rpm.	79
Figura 5.21. Longitudes de cuerda sin procesar de 100 rotaciones de un floculo virtual y la misma tras someterla a filtrado y promediado	80
Figura 5.22. Longitudes de cuerda de floculos de MDF 307 μm y distintas dimensiones fractales	81
Figura 5.23. D_f estimada frente a consignada	82
Figura 5.24. D_f estimada a partir de datos de DTC de suspensiones reales y la estimada por comparación para las velocidades de agitación consideradas.	83

ABREVIATURAS

RFR	Bosques aleatorios de regresión: (del inglés, <i>random forest regression</i>)
CDAMC	Celulosa modificada con dialdehído catiónica
DAMC	Celulosa modificada con dialdehído
CB	Concentración de biomasa
CCA	Crecimiento agregado-agregado (del inglés <i>cluster-cluster aggregation</i>)
DQO	Demanda química de oxígeno
D2	Dimensión fractal en dos dimensiones
Df	Dimensión fractal
DTC	Distribución de tamaños de cuerda
ECM	Error cuadrático medio
FBR	Fotobiorreactores
LMC	Longitud media de cuerda
SVR	Máquinas de vectores de soporte para regresión (del inglés, <i>support vector regression</i>)
MDF	Máximo diámetro de Feret
FBRM	Medida por reflexión de haz de láser enfocado (del inglés, <i>focused beam reflectance measurement</i>)
MLCB	Microscopía confocal láser de barrido
CNC-HC	Celulosa nanocristalina <i>hair</i> cationizada
PIE	Punto isoelectrico
RECM	Raíz del error cuadrático medio
TAP	Tris-acetato-fosfato (del inglés, <i>Tris-Acetate-Phosphate</i>)
NTU	Unidades nefelométricas de turbidez (del inglés, <i>nephelometric turbidity unit</i>)

RESUMEN EXTENDIDO

El incremento de la población mundial junto a la mejora de los medios de vida de los países en vías de desarrollo está causando un aumento de la presión sobre los recursos naturales del planeta. Esta situación ha propiciado que los gobiernos, el mundo académico y la industria hagan una búsqueda activa de nuevas fuentes de suministro que permitan el sostenimiento de la población y el desarrollo de la calidad de vida de la sociedad. A esta tendencia hay que añadir el creciente compromiso social por la conservación del medioambiente y el desarrollo sostenible. Se plantea pues, una coyuntura en la que coexisten una realidad demográfica y unos intereses sociales que son, a primera vista, contrapuestos. En este escenario, la producción sostenible de biomasa de microalgas se ha valorado como una opción atractiva para contribuir a resolver estos retos. La biomasa de microalgas como fuente de productos de alimentación o de productos químicos de interés presenta una serie de ventajas respecto a otras opciones. Por un lado, su producción puede llevarse a cabo en terrenos no aptos para el cultivo o la ganadería o incluso en el mar. Por otro lado, la posibilidad de emplear residuos ricos en nutrientes de otros procesos productivos, especialmente cuando la biomasa se destina a la producción de sustancias no destinadas al consumo humano, permitirían a la sociedad avanzar hacia la economía circular. Además, dado que las microalgas son seres fotosintéticos, su producción permitiría el aprovechamiento del CO₂ generado en procesos de combustión, actuando la industria productora de biomasa de microalgas como sumidero de carbono.

Con todo, el cultivo masivo de microalgas para la obtención de los productos anteriormente mencionados no ha llegado a materializarse por varios factores. Aparte de los aspectos regulatorios, la causa principal de este hecho hay que buscarla en los elevados costes de producción de la biomasa de microalgas, principalmente asociados al cultivo, en particular, al medio de cultivo, y al proceso de concentración de la biomasa.

Las estrategias que se han planteado para la disminución de los costes vinculados al cultivo de microalgas pasan por la reutilización del medio de cultivo en las instalaciones de producción, el empleo de aguas residuales tratadas como medio y el uso de sistemas de cultivo abiertos en los que se favorece el crecimiento de varias especies de microalgas o incluso de consorcios de microalgas con bacterias u hongos. Al plantearse estos esquemas productivos surgen una serie de problemas, entre los que se encuentra la dificultad de determinar la concentración de microalgas en el medio de forma rápida y robusta, en parte debido a que estos medios presentan una turbidez significativa o porque dicha turbidez varía con la edad del cultivo o a lo largo de las etapas de reutilización del medio.

En lo relativo a la reducción de los costes de concentración de la biomasa de microalgas, la estrategia que está recibiendo un mayor interés conlleva la implementación de una etapa de pre-concentración que permita reducir los costes energéticos vinculados a las etapas convencionales de concentración, tales como la filtración convencional, la filtración a vacío o la centrifugación. Como etapa de pre-concentración, la coagulación/floculación seguida o no de sedimentación, está considerada, a día de hoy, como la alternativa más atractiva desde el punto de vista de los costes relativamente bajos de inversión y de operación que esta técnica implica. En la etapa de coagulación/floculación, la selección de los agentes promotores de

agregación está condicionada por el destino final que tiene la biomasa producida, si bien se tiende, cada día más, a emplear productos obtenidos de forma sostenible y cuya gestión posterior no presente una problemática ambiental ni ocasione costes adicionales de tratamiento.

La operación de coagulación/floculación se optimiza al objeto de garantizar la clarificación de la suspensión que se trata, de forma que las partículas en suspensión se agreguen en flóculos que puedan separarse del medio acuoso de forma efectiva aguas abajo. El sistema de control del proceso suele operar actuando sobre la dosis de los agentes coagulantes y/o floculantes en función de una medida de la turbidez del medio, y también sobre el pH y la conductividad a través de un sistema de corrección de los mismos. Sin embargo, dado que las microalgas son microorganismos sujetos a ritmos circadianos y muy sensibles a pequeñas variaciones en su medio de cultivo, las características de los flóculos inducidos en la operación de coagulación/floculación pueden variar sin que se produzcan cambios apreciables en los valores de los parámetros de control. Este aspecto, si bien no necesariamente compromete la capacidad de clarificación del medio que tiene el proceso, puede provocar la pérdida de efectividad de las operaciones de separación subsiguientes que son muy sensibles al tamaño y morfología de los agregados.

En la actualidad, la mayoría de los métodos de caracterización de la morfología de los agregados de microalgas se ciñen a la estimación de la dimensión fractal proyectada por análisis de imagen, y los sistemas de estimación de la dimensión fractal en tres dimensiones están limitados, entre otros factores, por la concentración de las suspensiones de microalgas. No existen, en la actualidad, métodos que permitan estimar la dimensión fractal media en tres dimensiones de suspensiones de microalgas de forma robusta.

El principal objetivo de la presente tesis doctoral es el desarrollo de nuevos métodos de caracterización de suspensiones de microalgas mediante la aplicación de sistemas computacionales a los datos de distribución de tamaños o a imágenes microscópicas de dichas suspensiones. Los sistemas desarrollados se enmarcan en dos categorías principales, una de métodos de estimación de la concentración de biomasa seca de microalgas y otra de métodos para la estimación de la dimensión fractal media de los agregados de suspensiones de microalgas. Como objetivo secundario, se estudió el empleo de un nuevo agente floculante, una nanocelulosa cristalina catiónica, en la floculación de biomasa de microalgas.

Los sistemas de estimación de la concentración de biomasa emplean datos de distribución de tamaño de cuerda adquiridos por medio de una sonda de reflexión de haz de láser enfocado. Este método puede aportar datos prácticamente en tiempo real de la concentración de biomasa, con lo que resultaría útil en la monitorización de cultivos de microalgas en los que la turbidez del medio es apreciable y/o variable. Este sería el caso, por ejemplo, de los cultivos que empleen como medio de crecimiento aguas residuales tratadas o aquellos en los que se reutiliza el medio.

El primero de los métodos de estimación de la concentración de biomasa de microalgas se desarrolló para la caracterización de suspensiones de *Chlamydomonas reinhardtii*. El método se concibió para que la estimación de la concentración se obtuviese por correlación con los datos de tamaño de cuerda de las suspensiones empleando una red neuronal artificial tipo

perceptrón, cuya arquitectura se optimizó por medio de un algoritmo de recocido simulado. Con este método se consiguió un error cuadrático medio de predicción de $0,031 \text{ (g L}^{-1}\text{)}^2$ con la población de datos de validación.

El concepto aplicado para el desarrollo del sistema de estimación de la concentración de biomasa de *C. reinhardtii* se aplicó a otra cepa, *Chlorella sorokiniana*, y en la correlación se emplearon otros métodos de aprendizaje automático, en concreto modelos de regresión de bosques aleatorios y máquinas de vectores de soporte para la regresión. Estas condiciones se aplicaron al objeto de verificar el enfoque anterior con otra cepa de células más pequeñas (las células de *C. sorokiniana* son unas tres veces más pequeñas que las de *C. reinhardtii*). También se consideraron otros sistemas de aprendizaje automático especialmente apropiados para casos en los que el número de variables a emplear como predictores es elevado. En estos casos, los errores cuadráticos medios de predicción obtenidos fueron de $0,087 \text{ (g L}^{-1}\text{)}^2$ y $0,081 \text{ (g L}^{-1}\text{)}^2$ respectivamente, si bien en este caso, aplicando una validación cruzada en la que todos los datos de la población se emplean como datos de validación.

Los sistemas de estimación de la dimensión fractal tridimensional que se desarrollaron se encuadran, a su vez, en dos tipos, uno de estimación del parámetro por comparación de dimensiones fractales proyectadas reales y simuladas, y otro por interpretación de datos de tamaño de cuerda. El segundo método, que se puede implementar en línea y en tiempo real, podría emplearse para desarrollar sistemas de control de la floculación que garanticen la agregación de las partículas del medio en flóculos de morfología óptima para la separación de los mismos en posteriores etapas de concentración.

El sistema de estimación de la dimensión fractal por comparación consistió en la generación de agregados virtuales de tamaño y dimensión fractal conocida y posterior comparación de su proyección en el plano con las imágenes de flóculos reales tomadas por microscopía láser confocal de barrido. Con este método se obtuvieron resultados de estimación de la dimensión fractal compatibles con los de la literatura.

El sistema de software para la generación, rotación y proyección de flóculos virtuales desarrollado para la implementación del método anterior, sirvió como plataforma para la creación del sistema de estimación de la dimensión fractal a partir de datos de tamaño de cuerda. En este desarrollo se emplearon datos de tamaños de cuerda de flóculos virtuales para el entrenamiento y optimización de un modelo de regresión de bosques aleatorios que, posteriormente, se validó con datos reales. Los datos reales de validación se obtuvieron mediante la sonda de medida por reflexión de haz enfocado aplicada a suspensiones de microalgas de distinta concentración floculadas en las condiciones experimentales aplicadas en la creación del sistema de estimación de la dimensión fractal por comparación de proyecciones. El método de estimación por tamaños de cuerda fue capaz de estimar la dimensión fractal media de suspensiones de microalgas con independencia de su concentración, proporcionando resultados muy similares a los aportados por el método de comparación, especialmente en condiciones de agitación medias, consiguiéndose una discrepancia media respecto a los valores de referencia de 0,15.

La tesis se presenta como compendio de artículos, que se relacionan con lo descrito en este resumen de la manera siguiente. Los artículos I y II describen el desarrollo de modelos de

aprendizaje automático concebidos para estimar la biomasa seca de microalgas en distintos estado de agregación a través de la interpretación de los datos de tamaños de cuerda de las suspensiones correspondientes, estos últimos adquiridos a través de una sonda de medida por reflexión de haz enfocado. En el artículo III se estudia la aplicación de una nanocelulosa cristalina catiónica como agente floculante en suspensiones de microalgas, se proponen modelos para describir el proceso de agregación y se evalúa el efecto de la geometría de los agregados en la filtrabilidad de las suspensiones. El artículo IV describe el desarrollo de una metodología para estimar la dimensión fractal media de suspensiones de *C. sorokiniana* por medio de la comparación entre imágenes de flóculos tomadas por microscopía láser confocal y las proyecciones de flóculos virtuales de dimensión fractal conocida generados por ordenador. El artículo V aborda el desarrollo de un modelo de regresión de aprendizaje automático para estimar la dimensión fractal media de suspensiones de flóculos de microalgas por medio de la interpretación de los datos de tamaño de cuerda de las mismas tomados a través de la sonda de medida por reflexión de haz enfocado.

Por último, los conocimientos generados en la presente tesis se han aplicado a campos de la investigación distintos del que se aborda en este trabajo. En concreto, la optimización de los hiperparámetros y aplicación de modelos de máquinas de vectores de soporte para la regresión se aplicaron en el artículo *Morphological analysis of pulps from orange tree trimmings and its relation to mechanical properties* (Aguado et al. 2016), y el sistema de estimación directa de la dimensión fractal en el artículo titulado *Hairy cationic nanocrystalline cellulose as a novel flocculant of clay* (Campano et al. 2019, publicado en *Journal of Colloid and Interface Science*). Asimismo, los conocimientos generales sobre aprendizaje automático se han aplicado en el artículo titulado *Toward a description of the nanocrystalline cellulose aggregation state: correlating image analysis and Dynamic Light Scattering data* (Campano et al., enviado a *Langmuir*).

EXTENDED SUMMARY

The increase in the world population together with the improvement of the livelihoods of the developing countries is causing an increment in the pressure on the planet's natural resources. This situation has led governments, industry and academia to actively search for new resources to sustain the population and develop the quality of life of society. To this trend we must add the growing social commitment toward the environmental preservation and sustainable development. Therefore, a situation arises in which a demographic reality coexists with social interests that are, at first sight, opposed. In this scenario, the sustainable production of microalgae biomass has been valued as an attractive option to contribute to address these challenges. The biomass of microalgae as a source of food or chemical products of interest has a series of advantages over other options. On the one hand, its production can be carried out on land not suitable for cultivation or livestock, and even at sea. Additionally, the possibility of using nutrient-rich waste from other production processes, especially when biomass is used for the production of substances not intended for human consumption, would allow society to progress towards the circular economy. Furthermore, given that microalgae are photosynthetic beings, their production would allow the use of CO₂ generated in combustion processes, acting the microalgal production industry as a carbon sink.

Nevertheless, the mass cultivation of microalgae to obtain the aforementioned products has not been materialized due to several factors. Apart from regulatory aspects, the main cause of this fact must be sought in the high production costs of microalgae biomass, mainly associated with the cultivation, in particular to the growing medium, and in the biomass concentration process.

The strategies proposed for the reduction of the costs related to the cultivation of microalgae go through the reuse of the culture medium in the production facilities, the use of treated wastewater as a medium and the use of open farming systems in which the growth of several species of microalgae or even consortia of microalgae with bacteria or fungi are promoted. When considering these production schemes a series of problems arise, among which is the difficulty to determine the microalgal concentration in the medium in a quick and robust manner, partly because such media present a significant turbidity or because said turbidity varies with the age of the culture or throughout the stages of medium reuse.

As for the reduction of the concentration costs of the microalgae biomass, the strategy drawing a greater interest involves the implementation of a pre-concentration stage that allows to reduce the energy costs linked to the conventional concentration stages, such as conventional filtration, vacuum filtration or centrifugation. As a pre-concentration stage, coagulation / flocculation, followed or not by sedimentation, is considered, today, as the most attractive alternative owing to the relatively low investment and operation costs involved in this operation. In the coagulation / flocculation stage, the selection of aggregation promoting agents is determined by the final use of the biomass produced, although the present trend is to use products obtained in a sustainable manner and whose subsequent management does not entail environmental problems or incur additional treatment costs.

Coagulation / flocculation operations are normally optimized in order to ensure clarification of the suspension being treated, so that the particles in suspension are aggregated in flocs that can be effectively separated downstream from the aqueous medium. The process control system usually operates by acting on the dose of the coagulating or flocculating agents depending on a measure of the turbidity of the medium, and also on the pH and conductivity through a system for correcting them. Nonetheless, since microalgae are microorganisms subject to circadian rhythms and very sensitive to small variations in their culture medium, the characteristics of the flocs induced in the coagulation / flocculation operation can vary without appreciable changes in the values of the control parameters. This aspect, although not necessarily compromising the medium clarification capacity of the process, can cause a loss of effectiveness of the subsequent separation operations that are very sensitive to floc size and morphology.

At present, most of the morphological characterization methods of microalgae aggregates are restricted to estimating the projected fractal dimension of flocs by image analysis, being the three dimensions fractal dimension estimation systems limited, among other factors, by the concentration of microalgal suspensions. Currently, there are no methods to estimate the average fractal dimension in three dimensions of suspensions of microalgae in a robust manner.

The main objective of the present doctoral thesis is the development of new methods of characterization of microalgal suspensions through the application of computational systems to the size distribution data or to microscopic images of said suspensions. The systems developed fall into two main categories, one of methods for estimating the dry biomass concentration of microalgae and, the other, of methods to estimate the average fractal dimension of aggregates in microalgal suspensions. As a secondary objective, the use of a new flocculating agent, a cationic crystalline nanocellulose was studied for the flocculation of microalgae biomass.

The systems developed to estimate the biomass concentration of microalgae employ chord length distribution data acquired by means of a focused beam reflectance probe. This method can provide near real-time data on the concentration of biomass, which would be useful in the monitoring of microalgae cultures in which the turbidity of the medium is appreciable and / or variable. This would be the case, for example, of crops that use treated wastewater as a growth medium or those in which the medium is reused.

The first method to estimate the concentration of microalgae biomass was developed for the characterization of suspensions of *Chlamydomonas reinhardtii*. The method was conceived so that the estimation of the concentration was obtained by correlation with the chord length distribution data of the suspensions using a perceptron-type artificial neural network, whose architecture was optimized by means of a simulated annealing algorithm. With this method, a mean square error of prediction of 0.031 (g L⁻¹)² was achieved with the population of validation data.

The concept applied to the development of the biomass concentration estimation system of *C. reinhardtii* was applied to another strain, *Chlorella sorokiniana*, and in the correlation, other machine learning methods were used, namely random forest regression models and of

support vectors machines for regression. These conditions were applied for the purpose of verifying the previous approach with another strain of smaller cells (*C. sorokiniana* cells are about three times smaller than those of *C. reinhardtii*) and with other machine learning systems especially appropriate for cases in the that the number of variables to be used as predictors is high. In these cases, the mean squared prediction errors obtained were $0.087 \text{ (g L}^{-1}\text{)}^2$ and $0.081 \text{ (g L}^{-1}\text{)}^2$ respectively, although in this case, applying a cross-validation in which all the data of the population are used as validation data.

The systems to estimate the three dimensional fractal dimension that were developed according to two different approaches. In one, the estimation of the parameter is done by comparing the projected fractal dimension of real and simulated flocs and, in the other, by interpreting the chord length distribution data of microalgal suspensions. The second method, which could be implemented online and in real time, could be employed to develop flocculation control systems that guarantee the aggregation of particles in flocs of optimum morphology to achieve their separation in ensuing concentration stages.

The estimation system of the fractal dimension by comparison consisted in the generation of virtual aggregates of known size and fractal dimension and later comparison of its projection in the plane with the images of real flocs taken by confocal laser scanning microscopy. With this method, fractal dimension estimation results compatible with those of the literature were obtained.

The software system for the generation, rotation and projection of virtual flocs developed for the implementation of the previous method, served as a platform for the creation of the fractal dimension estimation system by interpreting chord length distribution data. In this development, chord length data of virtual were used for training and optimization of a random forest regression model that was subsequently validated with real data. The real validation data were obtained by means of the probe of measurement by focused beam reflectance applied to suspensions of microalgae of different concentrations flocculated in the experimental conditions applied in the creation of the system to estimate fractal dimension by comparison. The chord length distribution-based estimation method was able to estimate the average fractal dimension of microalgal suspensions regardless of their concentration, giving very similar results to those provided by the comparison method, especially under medium agitation conditions, achieving an average discrepancy with respect to the reference values of 0.15.

This doctoral thesis is presented as a compendium of articles, being these related to the contents of this summary as follows. Articles I and II describe the development of machine learning models designed to estimate the dry biomass of microalgae in different state of aggregation through the interpretation of the rope size data of the corresponding suspensions acquired through a measurement probe by focused beam reflection. In Article III the application of a cationic crystalline nanocellulose as a flocculating agent in suspensions of microalgae is studied, models to describe the aggregation process are proposed and the effect of the geometry of the aggregates on the filterability of the suspensions is evaluated. Article IV describes the development of a methodology to estimate the average fractal dimension of suspensions of *C. sorokiniana* by means of the comparison between images of flocs taken by confocal laser microscopy and the projections of computer generated virtual flocs of known

fractal dimension. Article V addresses the development of a machine learning regression model to estimate the average fractal dimension of suspensions of microalgae flocs by interpreting the chord length distribution data acquired through a focused beam reflectance measurement probe.

Finally, the knowledge generated in this thesis has been applied to research fields other than the one addressed in this paper. In particular, the optimization of the hyperparameters and the application of models of support vector machines for the regression were applied in the article Morphological analysis of pulp from orange tree trimmings and its relation to mechanical properties (Aguado et al., 2016), and direct estimation system of the fractal dimension in the article entitled *Hairy cationic nanocrystalline cellulose as a novel flocculant of clay clay* (Campano et al. 2019, published in *Journal of Colloid and Interface Science*). Likewise, the general knowledge on machine learning acquired has been applied in the article titled *Toward a description of the nanocrystalline cellulose aggregation state: correlating image analysis and Dynamic Light Scattering data* (Campano et al., submitted to *Langmuir*).

1

INTRODUCCIÓN

1.1. LAS MICROALGAS

Las microalgas son seres unicelulares eucarióticos en su gran mayoría fotoautotróficos con alta eficiencia fotosintética. Las microalgas son consideradas un grupo distintivo de microorganismos (Pulz and Gross, 2004). En virtud a su capacidad para transformar la energía lumínica en energía química, las microalgas, junto con las algas, son los productores primarios más importantes en las cadenas tróficas acuáticas, tanto en ecosistemas de agua dulce como en los salinos en todas las áreas climáticas del planeta.

Si bien el mecanismo de fotosíntesis de las microalgas es similar al de las plantas superiores, en la mayoría de los casos, la eficiencia en la conversión a biomasa de la energía solar es superior en las microalgas (Abdullah et al., 2017). La entidad de las microalgas en el ecosistema global queda patente si se tiene en cuenta que estos microorganismos se estima que son responsables de aproximadamente el 50% de la fijación de carbono orgánico global (Ratha and Prasanna, 2012).

Se cree que la variedad de microalgas es enorme, estimándose que pueden existir entre 200.000 y varios millones de especies de microorganismos en este grupo (Norton et al., 1996). La magnitud de estas cifras se pone de relieve por comparación con el número de especies de plantas superiores que se estima que existen en la actualidad y que estaría comprendido entre las 250,000 (Rieseberg, 1997) y las 320.000 (Prance et al., 2000).

Las microalgas con usos industriales conforman una gran familia de microorganismos productores primarios de biomasa. En general, las especies explotadas comercialmente se han dedicado a la producción de alimento vivo para acuicultura, alimentos funcionales, aditivos para piensos, productos químicos de alto valor en *nutracéuticos*, cosméticos y reactivos químicos (Liu et al., 2016; Priyadarshani and Rath, 2012). En los últimos años, las microalgas

han cobrado, además, cierta relevancia como elemento vehicular en la producción industrial de biocombustibles (Maeda et al., 2018) y en el desarrollo de sistemas sostenibles de tratamiento de aguas (Udaiyappan et al., 2017). Las microalgas constituyen una fuente potencial de sustancias de utilidad industrial. La tabla 1.1 recoge la composición en proteínas, carbohidratos y lípidos de algunas cepas de microalgas (Guccione et al., 2014; Illman et al., 2000; Mobin and Alam, 2017; Priyadarshani and Rath, 2012).

Tabla 1.1. Contenido en proteínas, carbohidratos y lípidos de biomasa de algunas cepas de microalgas expresado como porcentaje de biomasa seca.

Cepa	Proteínas	Carbohidratos	Lípidos
<i>Anabaena cylindrica</i>	43–56	25–30	4–7
<i>Botryococcus braunii</i>	40	2	33
<i>Chlamydomonas reinhardtii</i>	48	17	21
<i>Chlorella pyrenoidosa</i>	57	26	2
<i>Chlorella vulgaris</i>	41–58	12–17	10–22
<i>Chlorella sorokiniana</i>	40–45	25–38	20–24
<i>Dunaliella bioculata</i>	49	4	8
<i>Dunaliella salina</i>	57	32	6
<i>Dunaliella tertiolecta</i>	29	14	11
<i>Euglena gracilis</i>	39–61	14–18	14–20
<i>Haematococcus pluvialis</i>	17–25	32–37	36–40
<i>Porphyridium cruentum</i>	28–39	40–57	9–14
<i>Prymnesium parvum</i>	28–45	25–33	22–39
<i>Scenedesmus dimorphus</i>	8–18	21–52	16–40
<i>Scenedesmus obliquus</i>	50–56	10–17	12–14
<i>Scenedesmus quadricauda</i>	47	–	1.9
<i>Spirogyra</i> sp.	6–20	33–64	11–21
<i>Spirulina maxima</i>	60–71	13–16	6–7
<i>Spirulina platensis</i>	42–63	8–14	4–11
<i>Synechococcus</i> sp.	63	15	11
<i>Tetraselmis maculata</i>	52	15	3

Como puede colegirse de la tabla 1.1, la composición de las microalgas es diferente en función de la especie, y también presenta diferencias dentro de una misma cepa, estas últimas atribuibles a las condiciones en las que se realiza su cultivo (Chiranjeevi and Mohan, 2016).

En esta tesis doctoral se han considerado dos especies de microalgas, *Chlamydomonas reinhardtii* y *Chlorella sorokiniana*. La selección de estas especies está condicionada por la actividad investigadora del Grupo de Investigación de Celulosa y Papel, que ha realizado estudios de floculación de cultivos de ambas especies en el marco de dos proyectos de investigación, en concreto el proyecto nacional CENIT VIDA, *Valorización Integral De Algas* (*C. reinhardtii*) y el proyecto con ref. 280756 del VII Programa Marco de la Comisión Europea titulado *E4WATER, Economically and Ecologically Efficient Water Management in the European Chemical Industry*. Las investigaciones desarrolladas dentro de la presente tesis, extienden y complementan los trabajos desarrollados en ambos proyectos en lo referente a la monitorización de las suspensiones floculadas de cepas cultivadas de las mencionadas especies.

La *C. reinhardtii* es una microalga clorófito de agua dulce de unos 10 µm de diámetro que cuenta con dos flagelo para desplazarse. Esta microalga se ha estudiado de forma intensiva

como modelo de organismo y en la actualidad está despertando mucha atención académica por su capacidad para la producción de hidrógeno (Noone et al., 2017), por su habilidad para nutrirse de fuentes de carbono orgánico en ausencia de luz y como plataforma para la producción de productos de alto valor recombinantes (Ramos-Martinez et al., 2017), siendo tal vez la microalga cuyo metabolismo se conoce más en profundidad (Fields et al., 2018).

La *C. sorokiniana* es un alga unicelular también clorófita y de agua dulce de unos 3 μm de diámetro. Esta microalga también puede crecer heterotróficamente y se caracteriza por exhibir unas tasas de crecimiento significativamente altas (Sorokin and Krauss, 1959) y por su capacidad para tolerar temperaturas elevadas (Cazzaniga et al., 2014), aspectos que la convierten en una buena candidata para el cultivo a gran escala (Cazzaniga et al., 2014) y el tratamiento de aguas residuales (Álvarez-Díaz et al., 2017), si bien el pequeño tamaño de sus células dificulta su separación.

1.1.1. Metabolitos de microalgas con potencial comercial

A diferencia de la biomasa procedente de las plantas, la biomasa de microalgas es pobre en lignina y muy rica en sustancias con valor comercial, tales como proteínas (40-70%), hidratos de carbono (12-30%), lípidos (4-20%), carotenos (8-14%) y vitaminas. Por su composición, las microalgas constituyen una alternativa muy atractiva para la producción de productos alimenticios destinados al consumo humano y animal, suplementos nutricionales, pigmentos, cosméticos, productos farmacéuticos y biocombustibles. Para dar una idea del potencial de estos microorganismos, en las últimas décadas se han identificado más de 15.000 nuevos compuestos en las microalgas (Cardozo et al., 2007). Entre los compuestos de mayor interés comercial que se pueden extraer de las microalgas cabe destacar los siguientes.

Ácidos grasos poliinsaturados

Los ácidos grasos poliinsaturados (AGP), conocidos por su acrónimo *PUFA* (en inglés *polyunsaturated fatty acids*) se denominan así por contar en su cadena hidrocarbonada con más de un enlace doble carbono-carbono. En esta familia de compuestos se encuentran los ácidos omega-3, en concreto el ácido eicosapentaenóico (EPA) y el ácido docosahexaenóico (DHA). Ambos compuestos son ácidos grasos esenciales, o lo que es lo mismo, no sintetizables por el ser humano y son de crucial importancia para el sistema nervioso y en la regulación de la respuesta inflamatoria. Las microalgas son los productores primarios de ácidos grasos poliinsaturados en la cadenas tróficas marinas (Devos et al., 2006). Existe una gran variedad de especies de microalgas con capacidad para producir cantidades significativas de AGP, si bien posiblemente las especies con mayor proyección comercial en esta línea serían los géneros *Nannochloropsis* and *Phaeodactylum* como fuentes de EPA y el género *Isochrysis* como fuente de DHA (Ryckebosch et al., 2012).

Esteroles

Los esteroides son moléculas lipídicas policíclicas con unas funciones muy importantes en la fisiología de las células eucarióticas siendo componentes esenciales de la membrana celular e interviniendo, entre otros, en los procesos de reproducción celular y transducción de señales y en la modulación de la actividad de las enzimas unidas a la membrana. Ciertos esteroides presentes en las microalgas son capaces de inducir una disminución en los niveles de

colesterol en sangre (Devaraj et al., 2004), un efecto antiinflamatorio, y son precursores esenciales en la síntesis de vitaminas (Hamed et al., 2015).

Carotenoides

Los carotenoides, en su gran mayoría, son compuestos de la familia de los tetraterpenos, caracterizados pues por estar constituidos por 8 unidades de isopreno. Los carotenoides son compuestos con una actividad vital en las microalgas, pues están asociados a la captura y transmisión de la radiación lumínica y a la protección del centro de reacción fotosintético de la oxidación. En los animales, los carotenoides tienen actividad antioxidante, antiinflamatoria, antitumoral y son precursores de vitaminas como el retinol (Guedes et al., 2011). En acuicultura, los carotenoides se emplean para aportar pigmentación a crustáceos, como el camarón o la langosta, y a pescados de cultivo, como la trucha y el salmón.

Entre los carotenoides más relevantes producidos por las microalgas están el β -caroteno producido por la microalga halotolerante *Dunaliella salina* (Zhang et al., 2014) o la astaxantina, identificada como uno de los compuestos más potentes contra el estrés oxidativo que se conocen (Pashkow et al., 2008), que es producida principalmente por la microalga *Haematococcus pluvialis*.

Vitaminas

La biomasa de microalgas constituye una fuente de vitaminas, como por ejemplo, A, B1, B2, B6, B12, C, E, biotina, ácido fólico o ácido pantoténico (Gouveia et al., 2010). Ciertas especies de la *Chlorella* y de algas rojas (Rhodophyta) son conocidas por presentar altas concentraciones de cobalamina (vitamina B12) (de Jesus Raposo and de Moraes, 2015). La presencia y contenido de estos compuestos depende de la especie, de las condiciones de crecimiento de la biomasa y de la fase del ciclo de crecimiento en que se encuentre la misma.

Proteínas

Varias especies de microalgas son ricas en proteínas con un patrón de aminoácidos cuya biodisponibilidad es comparable a la de otras fuentes de proteínas, lo que apunta, a que estas especies podrían emplearse como fuente de proteína alimentaria (Hamed et al., 2015). Algunas especies presentan un contenido significativamente alto de proteínas, como el caso de especies de *Spirulina*, *Chlorella* e *Isochrysis*. Varias especies de *Spirulina* se han empleado desde antiguo como fuentes de proteínas por parte de civilizaciones como la azteca (Santley and Rose, 1979) o la maya y las comunidades de África Central que rodean al lago Chad siguen cosechando *Spirulina platensis* con fines alimentarios (Chen et al., 2016). Esta especie presenta un contenido proteico de entre el 60 y el 70%.

Polisacáridos

Dadas la relativamente alta velocidad de crecimiento de las microalgas cuando se controlan las condiciones de crecimiento, se considera que la mayor parte de polisacáridos producidos por microalgas podría tener aplicaciones industriales y comerciales.

Polisacáridos aislados de microalgas producidas comercialmente como la *Spirulina platensis* (LI et al., 2007a), la *Dunaliella salina* (Murthy et al., 2005) o la *Isochrysis galbana* (Sun et al., 2014) han mostrado propiedades antioxidantes muy notables.

De especial interés son los polisacáridos de la pared celular de ciertas microalgas rojas (*Porphyridium sp.*), un exopolisacárido de galactano sulfatado (EPS) que presenta una alta viscosidad en soluciones de baja concentración, comparable a los alginatos y a la goma guar (Bernaerts et al., 2018). Los polisacáridos de este tipo, presentan, además, propiedades antiinflamatorias (Wijesekara et al., 2011), antivíricas (Arad et al., 2006) y antitumorales (Sakamoto et al., 2003).

1.1.2. Métodos de cultivo de microalgas

El sistema de cultivo de microalgas a emplear de cara a la explotación comercial de la biomasa, depende principalmente del destino que se va a dar a esta última (Acién et al., 2018). Los sistemas de cultivo comercial de microalgas se suelen clasificar en dos categorías principales, los sistemas abiertos y los sistemas cerrados.

Los sistemas abiertos son estanques artificiales que pueden contar con un sistema de mezcla o carecer de él (balsas). Los primeros pueden ser de forma circular y mezclado a través de un agitador de eje central, de canales con agitación por paletas (*raceways*) o inclinados (Borowitzka and Moheimani, 2013). Los sistemas abiertos requieren inversiones relativamente bajas, son fáciles de manejar y baratos de mantener. Por otro lado, presentan baja productividad y, al estar el cultivo en contacto directo con la atmósfera, son susceptibles de contaminación y no permiten un buen control de las condiciones de proceso.

Los sistemas cerrados de aplicación comercial son los fotobiorreactores (FBR), en los que el cultivo no está en contacto directo con la atmósfera del entorno. El aislamiento del cultivo permite, por un lado, controlar las condiciones de operación de forma que se maximice la productividad del sistema, y por otro, evitar contaminaciones de otros microorganismos. Los fotobiorreactores se diseñan en base a dos configuraciones básicas, la tubular y la plana. A partir de estas dos tipologías, existe un gran número de configuraciones derivadas con geometrías, orientaciones y sistemas de circulación muy variadas (Liao et al., 2018).

Además de la geometría del reactor, en los sistemas cerrados es de suma importancia la forma en que se consigue la impulsión o agitación del cultivo y la manera en la que se favorece el intercambio de gases necesario para permitir un crecimiento y mantenimiento de la biomasa óptimos (Wang et al., 2012).

Los principales tipos de fotobiorreactores son los siguientes:

Columnas verticales

Fotobiorreactores tubulares orientados verticalmente en los que la agitación y el mezclado se consiguen por inyección de aire enriquecido en CO₂ por la base de la columna. Suelen tener hasta 0,2 m de diámetro y hasta 4 m de altura. Presentan altos coeficientes volumétricos de transferencia de gases, permitiendo un buen suministro de CO₂ y eliminación de O₂ (que puede inhibir la fotosíntesis), con lo que el crecimiento de las microalgas suele estar limitado por otros parámetros como la temperatura o la intensidad de luz (Wang et al., 2012).

Fotobiorreactores de placas planas

Son reactores en los que el cultivo se mantiene en una lámina relativamente fina por medio de dos superficies paralelas de material transparente o translúcido, lo que se traduce en una elevada relación de superficie a volumen. Tal configuración facilita enormemente la iluminación del cultivo. La orientación de los paneles en estos reactores puede ser vertical o inclinada al objeto de favorecer la captación de luz solar. Estos sistemas presentan ciertas limitaciones asociadas a la deposición de células sobre la superficie del material contenedor del medio. Los paneles de captación de este tipo de reactores se suelen fabricar de plástico rígido o flexible (bolsas). En general los paneles de plástico rígido suelen tener unas dimensiones que eviten el empleo de materiales de alta resistencia mecánica (Acién Fernández et al., 2013).

Fotobiorreactores tubulares.

Este tipo de reactor es el sistema de cultivo cerrado más popular. Los FBR tubulares están constituidos por tubos transparentes dentro de los cuales se hace circular el cultivo. Los tubos suelen tener un diámetro menor que 0,1 m para asegurar la transferencia de luz al medio de cultivo (Wang et al., 2012). Los reactores se construyen de acuerdo a una amplia variedad de diseños, pudiendo ser rectos, curvados, en espiral o mixtos. Como en el caso anterior, pueden encontrarse en orientaciones vertical, horizontal e inclinada. La circulación del cultivo en los tubos se consigue por bombeo directo o a través de un sistema *airlift*. Al objeto de eliminar el oxígeno producido en la fotosíntesis, se pasa el cultivo a través de una columna de burbujeo, donde se suministra aire y CO₂ y se controla la temperatura del medio (Acién Fernández et al., 2013). Este tipo de FBR permite un alto grado de control del proceso y una buena productividad. Las desventajas que presenta este tipo de sistemas están asociadas a la necesidad de aportar la necesaria turbulencia en el interior de los tubos que garantice la iluminación adecuada del cultivo y a la necesidad de evitar la acumulación de oxígeno disuelto, que a su vez impone limitaciones a la longitud máxima de los tubos. La mayor parte de la producción comercial a gran escala de microalgas se lleva a cabo en reactores tubulares (Borowitzka, 2018).

1.1.3. Cosechado de microalgas

El cosechado de microalgas es el proceso o procesos conducentes a la obtención de una pasta de microalgas con un contenido de agua reducido. El cosechado de microalgas, no comportando grandes dificultades a nivel técnico, constituye un reto de ingeniería a la hora de abordar una producción comercialmente viable de esta biomasa (Christenson and Sims, 2011) dada la baja concentración de biomasa que se alcanza en los FBR, el pequeño tamaño de las células de microalgas, entre 2 y 30 µm (Li et al., 2008; Zhang et al., 2010b), y su densidad, similar a la del agua.

El cosechado se suele llevar a cabo en una o dos etapas (Brennan and Owende, 2010). La aplicación de un enfoque u otro va a depender de los requisitos del producto final y del precio de la biomasa. En términos generales, el cosechado en dos etapas es de aplicación más común. En este caso se contempla una etapa de pre-concentración o cosechado primario (*thickening* empleando la terminología anglosajona) y otra de cosechado secundario (*dewatering*).

Cuando el cosechado se realiza en dos etapas, en la primera se parte de una suspensión diluida de microalgas, típicamente con una concentración entre 0.02 y 0.06 % de sólidos totales en suspensión, y se llega a una suspensión con una concentración de entre el 2 y el 7 % de sólidos totales. De esta suspensión intermedia se llega a una pasta con una concentración de entre el 5 y el 25 % de sólidos (Uduman et al., 2010a). Una revisión bastante exhaustiva de las técnicas de concentración de microalgas puede encontrarse en Barros et al. (2015). Las principales técnicas de cosechado primario de microalgas se resumen a continuación.

Floculación/coagulación

La floculación/coagulación es el proceso por el cual se consigue desestabilizar una suspensión por medio de la adición de compuestos químicos que promueven la agregación de las partículas desestabilizadas, que posteriormente se separan del medio clarificado obteniéndose una suspensión más concentrada que la de partida. La coagulación implica la neutralización de las cargas superficiales de las partículas en suspensión que formarían micro-flóculos, y la floculación consiste en la agregación de dichos agregados en otros de mayor tamaño. El proceso de agregación se puede llevar a cabo con sistemas dobles, es decir, aplicando coagulantes y floculantes, o en una sola etapa, normalmente empleando solo floculantes. Como sustancias coagulantes suelen emplearse sales metálicas, p. ej., de hierro o aluminio, si bien, en muchos casos estos coagulantes no son compatibles con el uso final de las microalgas y pueden impedir la reutilización del medio clarificado en el cultivo. En cuanto a los floculantes, se suelen emplear polímeros naturales o sintéticos, normalmente catiónicos en virtud a su capacidad de reducir o neutralizar la carga superficial negativa de las microalgas. El empleo de polímeros catiónicos como la poliacrilamida de alto peso molecular, el quitosano o el almidón cationizado se han empleado satisfactoriamente en la floculación de microalgas (Show and Lee, 2014), especialmente en especies de agua dulce. Como en el caso de los coagulantes, la elección del floculante a aplicar ha de tener en cuenta el destino final de la biomasa producida.

La floculación/coagulación es, a día de hoy, la técnica que más interés está suscitando como medio para mejorar la viabilidad económica de la producción de microalgas (Wan et al., 2015) por su versatilidad y bajo coste. La floculación/coagulación suele asociarse a una operación posterior de sedimentación.

Autofloculación

Consiste en inducir la floculación natural de las microalgas a través de un incremento del pH del medio, normalmente mediante la adición de hidróxidos de calcio, magnesio o sodio. Si bien se considera un proceso barato, su aplicación a nivel industrial no está extendida por la falta de control sobre el proceso de agregación, que depende de un elevado número de factores (Sirin, 2013). Además, el incremento de pH del medio limita la reutilización del medio de cultivo.

Biofloculación

La biofloculación es la agregación de las microalgas a través de los polisacáridos externos segregados por bacterias u hongos con las que se cultivan. La biofloculación se ha conseguido también cultivando conjuntamente cepas de microalgas auto-floculantes con otras que no

presentan esta capacidad (Salim et al., 2011). Si bien la biofloculación evita el empleo de floculantes que pueden resultar contaminantes para el cultivo, la biomasa agregada resultante presenta una contaminación biológica que puede ser incompatible con numerosas aplicaciones. Por otro lado, en ocasiones, el cultivo conjunto de microalgas y otros microorganismos puede requerir un sustrato suplementario como fuente de carbono para estos últimos, con el consiguiente encarecimiento del cultivo.

Sedimentación

La sedimentación es un proceso de concentración de bajo coste, aunque su aplicación directa para concentrar los cultivos de microalgas no se considera una opción viable a nivel industrial, principalmente por las bajas velocidades de sedimentación, entre 0.1 y 2.6 cm h⁻¹ debidas al pequeño tamaño de las partículas en suspensión del medio, y la posibilidad de que la biomasa sufra deterioros en el proceso (Eldridge et al., 2012). La sedimentación se suele aplicar, por lo tanto, en combinación con procesos de floculación/coagulación (Grima et al., 2003).

Flotación por aire disuelto

En la flotación se consigue la adhesión de las células del cultivo a un caudal de burbujas de aire, que las arrastra formando una espuma rica en la biomasa en la parte superior del equipo. La flotación se considera más eficiente que la sedimentación (Hanotu et al., 2012) cuando se aplica directamente al cultivo de microalgas, si bien, como en el caso anterior suele aplicarse en combinación con una etapa de floculación/coagulación. La flotación presenta una serie de ventajas interesantes, entre ellas, que se ha demostrado la viabilidad de su empleo a escala industrial para suspensiones de partículas minerales o en tratamientos de agua (Rubio et al., 2002) y que requiere de un espacio reducido para su instalación.

Técnicas alternativas

Existen otras técnicas para conseguir la preconcentración de la biomasa de microalgas, como son la electro-flotación y la electro-coagulación seguida de flotación. La electro-flotación es similar a la flotación por aire disuelto, si bien en la primera técnica las burbujas se generan a través de la electrolisis del agua. La electro-flotación puede considerarse una alternativa a la flotación en el caso de producciones a pequeña escala (Kyzas and Matis, 2016). La electro-coagulación seguida de flotación se basa en la producción de cationes metálicos in-situ a partir de la oxidación electrolítica de un ánodo metálico (Vandamme et al., 2011). Al igual que en el caso de la coagulación con sales metálicas, la biomasa floculada presenta contaminación por iones metálicos, si bien en este caso en un grado notablemente menor (Vandamme et al., 2013). La implantación de estos métodos es a día de hoy minoritaria debido a los costes de operación asociados que presentan (Christenson and Sims, 2011).

Procesos de cosechado secundario

Estos procesos suelen ser de tipo mecánico, siendo los más importantes la centrifugación y la filtración.

▪ Centrifugación

La centrifugación es efectiva en la separación de la biomasa preconcentrada de microalgas, y en algún caso también del cultivo sin preconcentrar. La aplicación de esta técnica tiene

asociada unos altos costes energéticos, que convierten a la centrifugación en la alternativa más cara para conseguir la concentración de la biomasa de microalgas (Barros et al., 2015). La centrifugación puede ser, por lo tanto, una alternativa viable cuando el destino final de la biomasa es la producción de sustancias de alto valor añadido, siendo su aplicación demasiado costosa para la producción de sustancias de bajo valor (Dassey and Theegala, 2013). En cualquier caso, la eficiencia económica de la centrifugación puede mejorarse si se combina con otras técnicas de cosechado primario, como la floculación-sedimentación (Fasaei et al., 2018; Milledge and Heaven, 2013).

▪ Filtración

Un buen número de estudios han demostrado la efectividad de la filtración en diferentes variantes como técnica de concentración de la biomasa de microalgas (Mathimani and Mallick, 2018; Singh and Patidar, 2018). Además, los consumos energéticos asociados a los procesos de filtración son relativamente bajos. El principal inconveniente que presenta esta alternativa de concentración estriba en la ocurrencia frecuente de fenómenos de ensuciamiento del medio filtrante (Milledge and Heaven, 2013). Al igual que en el caso de la centrifugación, se ha demostrado que la combinación de la filtración con otros procesos de preconcentración, especialmente, floculación y sedimentación, puede mejorar la eficiencia económica del proceso (Fasaei et al., 2018; Singh and Patidar, 2018).

1.1.4. Producción comercial de biomasa de microalgas

El cultivo a nivel comercial de biomasa de microalgas está orientado a la producción de sustancias de alto valor añadido, dado el elevado coste asociado a la producción de estos microorganismos (Suganya et al., 2016). Los costes de producción más bajos referidos por suministradores están en torno a los 5 € kg⁻¹ de biomasa seca de *D. salina* (cultivo en balsas), mientras que para algunas especies los costes pueden ser superiores a los 30 € kg⁻¹, como por ejemplo en el caso de la biomasa de *H. pluvialis* cultivada en fotobiorreactores tubulares (Borowitzka, 2018).

En la actualidad, la producción de biomasa de microalgas destinadas al consumo humano o animal se concentra principalmente en Asia, destinándose la mayor parte de la producción al mercado de los suplementos nutricionales y, de forma minoritaria, a la producción de productos alimenticios (Vigani et al., 2015). Si bien el uso de microalgas para la producción masiva de proteínas o carbohidratos dista de ser una realidad a día de hoy, la Unión Europea está orientando parte de sus recursos de investigación y desarrollo para que la producción industrial de microalgas llegue a materializarse (Enzing et al., 2014).

A día de hoy es posible encontrar empresas dedicadas a la producción de biomasa de microalgas destinada a la fabricación de biocombustibles (Maeda et al., 2018). La producción de estos últimos en términos absolutos es anecdótica en comparación con la de combustibles fósiles o la procedente de otros combustibles basados en la biomasa.

Sin tener en cuenta las barreras no tecnológicas, la viabilidad de la producción industrial de microalgas pasa por reducir los costes totales asociados a la obtención de este tipo de biomasa. En el caso de la producción de biocombustibles, se estima que los costes de producción

podrían reducirse con la utilización de aguas residuales tratadas (Raghuvanshi et al., 2018; Salama et al., 2017). En cualquier caso, el coste total de la obtención de microalgas, ya sea para fabricar biocombustibles como otros productos, puede abarataarse notablemente reduciendo el coste asociado al cosechado de la biomasa (Barros et al., 2015).

1.2. FLOCULACIÓN DE SUSPENSIONES DE MICROALGAS

Se estima que los costes de las etapas de cosechado (preconcentración y eliminación del agua) suponen entre un 20 y un 30% del coste total de ciclo productivo de la biomasa seca (Barros et al., 2015). Estos costes se atribuyen a varios factores, siendo los más relevantes el tamaño de la microalgas, del orden de unas pocas micras, las bajas concentraciones de biomasa que presentan los cultivos, entre 0,5 y 4 g·L⁻¹ en base de materia seca (Chisti, 2013) y la densidad de las microalgas, muy parecida a la del agua. Entre las alternativas tecnológicas para inducir la pre-concentración de la biomasa de microalgas, la coagulación-floculación combinada con una etapa de sedimentación, centrifugación o filtración está considerada como la técnica más económica de cuantas se consideran maduras (Barros et al., 2015; Choy et al., 2018). La floculación de microalgas conlleva la neutralización total o parcial de la carga superficial negativa que presentan las células en suspensión de estos microorganismos, con lo cual se favorece la formación de agregados que presentan una mayor tendencia a la sedimentación o permiten la aplicación de una etapa de filtración, centrifugación o flotación (Chatsungnoen and Chisti, 2016).

La agregación de suspensiones de microalgas se ha investigado aplicando diferentes métodos. Además de la floculación química, se ha estudiado una variedad de técnicas, como son la biofloculación por medio de la combinación de cepas de microalgas (Salim et al., 2011), de estas con bacterias (Lee et al., 2009) o con hongos filamentosos (Zhou et al., 2013), la electrocoagulación (Lee et al., 2013), o la floculación inducida por pH (Vandamme et al., 2012). A pesar de los avances logrados, la falta de un adecuado control durante los procesos de auto y biofloculación (Schenk et al., 2008) y la necesidad de emplear sustratos adicionales para el crecimiento bacteriano en el caso de la biofloculación (Lee et al., 2009) limitan el empleo de estas técnicas a escalas industriales.

La floculación química por medio de la adición de agentes externos al cultivo se sigue considerando la vía más apropiada para inducir la agregación de la biomasa de microalgas. La literatura que aborda este enfoque es muy abundante en referencias, habiéndose investigado una gran variedad de agentes con capacidad para inducir la floculación de las suspensiones. Se ha estudiado la agregación con sales inorgánicas, como las aluminio (AlCl₃, Al₂(SO₄)₃) o de hierro (FeCl₃, Fe₂(SO₄)₃), con polímeros sintéticos como la poliácridamida catiónica, el óxido de polietileno o el polyDADMAC y empleando compuestos naturales modificados o no, como almidón cationizado, quitosano o ácido poliglutámico (Roselet et al., 2015; Van Haver and Nayar, 2017).

La selección de los floculantes para llevar a cabo los ensayos de la presente investigación en los que se induce la agregación de suspensiones de microalgas se ha llevado a cabo de acuerdo a un criterio de sostenibilidad. Este enfoque, además de partir de una convicción personal, se alinea con los valores de la sociedad actual, que cada día más, demanda productos y

tecnologías de menor impacto medioambiental. En la mayor parte del trabajo se ha empleado quitosano y en menor medida celulosa nanocristalina *hairycationizada*. Estos dos floculantes, siendo de naturaleza distinta, presentan un rasgo común y es que ambos proceden de fuentes sostenibles: el quitosano se produce a partir de la quitina que forma el exoesqueleto de los artrópodos y está presente en los hongos, y los nanocristales se producen a partir de celulosa.

El quitosano

El quitosano es el producto que se obtiene de la desacetilación de la quitina en medio básico. La quitina es un biopolímero estructural, probablemente el segundo más abundante en la naturaleza después de la celulosa (Strand et al., 2005). La quitina se obtiene un su mayor parte del subproducto de conchas de crustáceos marinos que se genera en la industria conservera. La quitina es un polímero lineal que forma estructuras de microfibras cristalinas con propiedades estructurales en artrópodos y hongos. El proceso de obtención del quitosano es sencillo. Se lleva a cabo a partir de quitina previamente desproteinizada, desmineralizada y decolorada, normalmente por medios químicos. Esta quitina se somete a tratamiento alcalino con un 40 a 50% de NaOH (Hamed et al., 2016). En la figura 1.1 se muestra la estructura de la molécula de quitosano. Los tratamientos alcalinos hidrolizan los grupos acetilo (COCH_3) y transforman las unidades N-acetil-D-glucosamina en unidades de D-glucosamina con grupos amino libres ($-\text{NH}_2$), que confieren la carga positiva al quitosano.

Es posible producir quitosanos con diferentes grados de desacetilación variando la temperatura de reacción, el tiempo y la concentración de la solución alcalina (Teng, 2012).

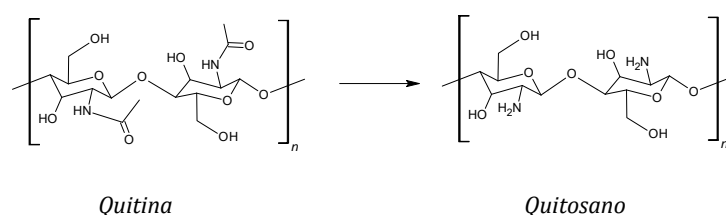


Figura 1.1 Estructuras moleculares de quitina y quitosano

El quitosano está considerado no tóxico, biocompatible y biodegradable (Anaya et al., 2013; Zhang et al., 2010a).

Celulosa nanocristalina *hairycationizada*

La celulosa nanocristalina *hairycationizada* (CNC-HC) se obtiene de la celulosa, el biopolímero más abundante de la naturaleza. El empleo de CNC-HC como floculante no se está descrito en la literatura en el momento de la redacción de la presente memoria. La investigación del uso de CNC-HC como floculante que se ha realizado en el marco de esta tesis es la segunda de la que el autor tiene conocimiento, habiéndose realizado la primera en el mismo grupo de investigación con suspensiones de caolín.

La preparación de CNC-HC a partir de pulpa de celulosa se describe por vez primera en 2016 por Yang y van de Ven. El proceso consta de dos pasos principales (Figura 1.2). En el primero, (Figura 1.2a) la pulpa se oxida con NaIO_4 para producir celulosa modificada con dialdehído (DAMC por sus siglas en inglés). En el segundo paso (Figura 1.2b) la carga de la DAMC se cambia a catiónica por una reacción de tipo Schiff con el reactivo de Girard T (GT),

produciendo DAMC catiónica (CDAMC). Finalmente, se aplica un tratamiento térmico para nanofibrillar la CDAMC y producir así CNC-HC.

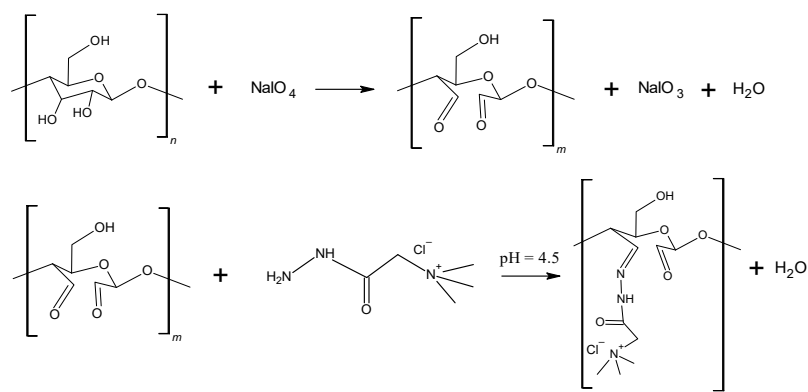


Figura 1.2. Producción de CNC-HC a partir de celulosa

Los CNC-HC se caracterizan por tener dos extremos amorfos cargados positivamente unidos a través de un eje cristalino común de 100 a 200 nm de longitud y pocos nanómetros de ancho. CNC-HC presenta una densidad de carga muy alta y su geometría particular y distribución de carga les confiere propiedades singulares de relevancia en los procesos de floculación, como su alta densidad de carga y su capacidad para inducir la agregación a través de diferentes vías de crecimiento en función de su concentración.

1.3. MORFOLOGÍA DE LOS AGREGADOS DE MICROALGAS

1.3.1. Los objetos fractales y los agregados de partículas

La floculación de una suspensión es el resultado de una serie de procesos complejos interconectados que transcurren de forma más o menos simultánea. Entre dichos fenómenos cabe mencionar las interacciones electrostáticas, la adsorción de moléculas de polímero en la superficie de las partículas, la redistribución (o reconformación) de cadenas poliméricas adsorbidas; colisiones entre partículas desestabilizadas para formar agregados (flóculos), la ruptura de los mismos y reconfiguración del agregado. La importancia de cada proceso depende de un amplio número de factores. De un lado, de las características del floculante en cuanto a su composición química, estructura, peso molecular, densidad de carga y concentración. En cuanto a las partículas a flocular, son determinantes de cara a la floculación aspectos como su naturaleza química, su distribución de tamaños, su geometría y su carga. Por otra parte, las características de la suspensión como la conductividad, pH, temperatura y la carga iónica van a condicionar la forma en que se produce la agregación. Por último, el tiempo de contacto y la intensidad de la agitación aplicada al sistema que se flocula influye también en el resultado del proceso de floculación y, por ende en la geometría de los agregados a obtener.

El primer trabajo en el que se abordó explícitamente la geometría fractal de los agregados de partículas de crecimiento aleatorio corresponde a Forrest and Witten Jr (1979). Los agregados formados por la agregación de partículas primarias a través de fenómenos de colisión

espontáneos o inducidos mediante agitación exhiben una naturaleza fractal (Jiang and Logan, 1991). Los agregados de microalgas, por extensión pueden considerarse objetos fractales.

El término fractal fue acuñado por el matemático de origen polaco y nacionalidad francesa Benoit Mandelbrot, que abordó la descripción de las estructuras fractales en su trabajo titulado *Les Objects Fractals: Forme, Hasard et Dimension* (Mandelbrot, 1975).

Un fractal es un objeto geométrico caracterizado por presentar autosimilitud, es decir, el objeto considerado como un todo es geoméricamente similar a las partes que lo constituyen con independencia del tamaño de las mismas. Los fractales son pues estructuras cuyas propiedades geométricas son invariantes cuando se cambia la escala con la que se realiza la medida. Los objetos fractales reales presentan una autosimilitud estadística (Berntson and Stoll, 1997).

Los primeros estudios relevantes acerca de la naturaleza fractal de los agregados formados a partir de partículas coloidales o sólidas comenzaron en la década de los años ochenta (Ball et al., 1987; Meakin, 1983).

La invariancia de la medida con la variación de escala puede expresarse por medio de la ecuación (1.1):

$$N \propto \varepsilon^{-D_f} \quad (1.1)$$

Donde ε es el factor de escala, D_f es la dimensión fractal y N es el número de unidades de medida resultantes del análisis del objeto fractal en el factor de escala considerado.

La expresión anterior puede adaptarse al caso de los objetos reales y, más concretamente, al de los agregados de partículas estableciendo como factor de escala una longitud característica del agregado que dará como resultado un cierto número de medidas que será proporcional a la masa del mismo si la densidad de las partículas primarias que lo forman tienen densidad constante. Esta relación de proporcionalidad podría expresarse como:

$$M \propto l^{-D_f}, \quad (1.2)$$

donde M es la masa del agregado, l es una medida de longitud característica del agregado y D_f es la dimensión fractal. D_f en la expresión anterior puede tomar valores entre 1, que correspondería a un agregado lineal de partículas primarias, y 3, valor que estaría asociado a un agregado compacto. La ecuación 1.2 ofrece información acerca de cómo se distribuyen las partículas primarias y, por tanto, la masa, en el agregado. Resulta pues patente, que la determinación de la dimensión fractal de los agregados puede aportar una información muy valiosa para caracterizar los agregados de partículas y predecir su comportamiento.

1.3.2. La morfología de los agregados de partículas en la separación sólido líquido

La geometría de las partículas en suspensión es un parámetro de capital importancia que condiciona el comportamiento de dichas partículas cuando estas se ven sometidas a procesos de interacción con el fluido en el que se encuentran, con otras partículas o con radiación electromagnética.

Las implicaciones de la forma en que se distribuyen las células de microalgas en los agregados son de suma importancia en el contexto de los sistemas de pre-concentración de biomasa, especialmente en aquellas técnicas consideradas de bajo coste, como son la sedimentación y la filtración.

Por un lado, en los procesos de sedimentación, la estructura geométrica de las partículas, junto con la longitud característica de las mismas, van a condicionar su velocidad de sedimentación.

Si la separación se lleva a cabo a través de un medio filtrante, la forma y el tamaño de los agregados determinarán las características de la torta de filtración y, con ella, la eficiencia del filtrado.

Por otro lado, cuando se aborda la monitorización de las suspensiones de sólidos por medios ópticos, la morfología de los agregados es un aspecto clave en los fenómenos de interacción entre la radiación electromagnética y la materia, siendo pues determinante en la eficacia y exactitud del método de medida. **La morfología de los agregados en los procesos de sedimentación**

Tradicionalmente, la velocidad terminal de las partículas en un fluido se ha relacionado con su longitud característica a través de la ley de Stokes (ecuación 1.3), que deriva la mencionada velocidad asumiendo el equilibrio de las fuerzas gravitacional, viscosa y de flotación para un agregado.

$$v_s^2 = \frac{4g(\rho_p - \rho_a)}{3\Omega c_D \rho_a} d, \quad (1.3)$$

donde v_s es la velocidad terminal del agregado, g la aceleración gravitatoria, ρ_p y ρ_a las densidades de la partícula y el agua, Ω es un factor de corrección para tener en cuenta la porosidad de las partículas y c_D es el coeficiente de arrastre.

La ley de Stokes en su planteamiento original solo ofrece una exactitud apreciable cuando se estima la velocidad terminal de partículas esféricas en fluidos con bajos números de Reynolds. Al objeto de computar el efecto de la geometría de las partículas sobre la velocidad terminal, se ha propuesto una amplia variedad de modificaciones a la ley de Stokes. En la mayor parte de los casos, la incorporación del término de forma al proceso de obtención de la velocidad terminal se ha hecho a través del coeficiente de arrastre, c_D .

El parámetro c_D se relaciona con el área transversal que la partícula proyecta al flujo del fluido y, en general, asume que este área es proporcional a una longitud característica L al cuadrado. Sin embargo, este no es el caso de las partículas porosas o de aquellas que presentan una estructura que resulta de la agregación de partículas primarias más pequeñas, que pueden presentar naturaleza fractal.

Tomando este hecho en consideración, en los últimos años se han llevado a cabo estudios que proponen modificaciones a las ecuaciones derivadas de la ley de Stokes que integran el efecto de la geometría de las partículas sobre la velocidad terminal a través de la dimensión fractal, como es el caso de la siguiente expresión (ecuación 1.4) que se ofrece como ejemplo y que propuso Vahedi and Gorczyca (2012).

$$v_s = \frac{1}{18} \theta g \frac{(\rho_p - \rho_a)}{\mu} d_{50}^{3-D_f} \frac{d^{D_f-1}}{1+0,15Re^{0,687}} \Phi, \quad (1.4)$$

Donde θ es el factor de forma adimensional del agregado, μ es la viscosidad dinámica del agua, d_{50} es la mediana de los tamaños de las partículas primarias que componen el agregado, D_f es la dimensión fractal en tres dimensiones del agregado, d es el diámetro del agregado, Re es el número de Reynolds y Φ un parámetro adimensional relacionado con la distribución de partículas primarias en el flóculo.

De las consideraciones anteriores se deduce que dos flóculos de la misma longitud característica (p. ej. máximo diámetro de Feret), que presenten distribuciones diferentes de masa y, por consiguiente, de partículas primarias, van a experimentar distintas velocidades terminales. Se puede inferir a su vez de esto, que cualquier tentativa de optimización de un proceso de floculación con sedimentación ha de concentrarse en establecer la combinación más adecuada entre tamaño de los agregados y su geometría de forma que se maximice la velocidad terminal de las mismas. De igual modo, un sistema de control de la floculación que pretenda tener un alcance exhaustivo ha de tener como objetivo la producción continua de agregados cuya combinación de tamaño y forma sea óptima para las etapas de separación subsiguientes. Dado que la geometría y tamaño de los agregados es función de las condiciones hidrodinámicas en las que se produce la floculación, un sistema ideal de control del proceso actuaría no solo sobre el pH y la dosis de agentes coagulantes y/o floculantes, sino también sobre la intensidad de la agitación que se aplica a la suspensión.

La morfología de los agregados en los procesos de filtración

La permeabilidad de la torta de filtración depende del tamaño de poro del medio filtrante y del binomio formado por el tamaño y la geometría de los flóculos que forman la torta. Se ha observado que la permeabilidad de la torta mejora cuanto menor es la dimensión fractal de los flóculos y mayor es el tamaño de los mismos (Cho et al., 2005; Lee et al., 2003). Así, si dos suspensiones de flóculos tienen el mismo tamaño medio, aquella cuyos agregados tengan una dimensión fractal media menor, serán más fácilmente filtrables.

Como en el caso de la sedimentación, se han propuesto modificaciones a la relaciones de Carman-Kozeny (Carman, 1937; Kozeny, 1927) para la estimación de la resistencia de la torta de filtración (Amjad et al., 2015). Un sistema capaz de monitorizar la geometría de los flóculos permitiría implementar un sistema de control de la floculación que garantizase que los agregados producidos tengan una dimensión fractal y un tamaño óptimos para que la filtración subsiguiente funcione de forma efectiva.

1.3.3. Caracterización de las suspensiones de sólidos

La monitorización de los sólidos en suspensión en matrices líquidas tiene como objetivo principal la cuantificación de la materia sólida por unidad de volumen de la matriz líquida considerada, bien de forma directa o a través de una variable indirecta, de carácter intensivo, que se le pueda relacionar. Además de la determinación de la concentración de materia sólida, en ocasiones se emplean técnicas de monitorización que permiten inferir el tamaño medio de las partículas en suspensión. La determinación de la geometría media de los agregados en suspensión, por otro lado, suele restringirse al ámbito de la investigación académica.

En el contexto de este trabajo, la concentración de sólidos es una medida para obtener información útil de cara al estado de la suspensión de biomasa y a los procesos a que se somete dicha suspensión. Por ejemplo, para establecer el estado de crecimiento de un cultivo de microalgas o la eficiencia de un proceso de separación de la biomasa del medio en que esta se cultiva.

En cuanto a la estimación del tamaño de los agregados en suspensión, su aplicación industrial en el ámbito que se aborda en esta tesis, se ha limitado al seguimiento puntual de los procesos de floculación. Cabe destacar que la estimación de tamaño sí se emplea en la monitorización de procesos industriales de cristalización, principalmente en la industria farmacéutica.

A continuación se escriben las principales técnicas de monitorización de suspensiones de sólidos.

1.3.3.1. Principales técnicas de monitorización de la concentración y tamaños de partícula en suspensiones de sólidos

Determinación de la concentración de sólidos por gravimetría

La determinación por gravimetría es la técnica convencional que se emplea para estimar la concentración de sólidos en suspensión. Es la técnica de referencia con cuyos resultados se relacionan las medidas indirectas de la concentración, que normalmente se basan en técnicas ópticas.

La media de la concentración de sólidos conlleva la toma de una muestra de volumen conocido y el procesamiento de la misma hasta la pesada de la fracción de sólidos contenida en el volumen inicial (Zhu and Lee, 1997). El procesamiento de la muestra implica normalmente la separación de la fracción sólida de la muestra por procesos físicos como la filtración o la centrifugación, a los que suele seguir una etapa de secado.

Esta técnica, si bien puede resultar bastante precisa, requiere de tiempos de procesamiento relativamente largos (Sonnleitner et al., 1992), especialmente cuando implica una etapa de secado, por lo que su aplicación para el seguimiento y control de procesos en los que intervienen los agregados de partículas es muy limitada.

Turbidimetría y nefelometría

La turbidimetría es una medida directa de la intensidad de luz transmitida a través de una muestra líquida. La materia en suspensión presente en el líquido provoca la dispersión de una parte de la luz incidente y puede absorber una parte de la energía del haz. La turbidimetría es adecuada para la caracterización de muestras en las que las partículas son relativamente grandes en comparación con la longitud de onda de la luz incidente empleada (Lambrou et al., 2009).

La nefelometría se define como la medida de la intensidad de luz dispersada a su paso por una muestra líquida. El detector de luz suele estar situado entre 15° y 90° de la trayectoria del haz incidente. La nefelometría se suele emplear en muestras poco concentradas que presentan partículas de tamaño pequeño.

La turbidez se correlaciona bien con la concentración de sólidos a bajas concentraciones pero ofrece resultados inexactos cuando la concentración supera un cierto límite, debido al efecto de solapamiento de partículas que se produce en el seno de la suspensión, que hace que la variación de la turbidez con la concentración deje de tener un comportamiento lineal. Al efecto de la concentración hay que sumar el efecto del tamaño y de la forma de las partículas en suspensión. Está demostrado que la forma en la que la masa sólida se distribuye en los agregados y la dimensión de los mismos ejercen un efecto sobre la turbidez del líquido en que están en suspensión (Druine et al., 2018). Se puede dar el caso pues, de que dos suspensiones de distinta concentración de sólidos y distinto estado de agregación produzcan un mismo valor de turbidez. Para la adecuada monitorización de suspensiones de microalgas a través la medida de la turbidez se requiere que se cumplan una serie de requisitos de concentración y estado de agregación de la biomasa, estando limitada su aplicación para el seguimiento de los procesos de separación por floculación.

Dispersión dinámica de luz

La dispersión dinámica de luz o espectroscopia de correlación fotónica es una técnica físico-química empleada para la estimación de la distribución de tamaños de partícula en suspensiones. La técnica consiste en correlacionar las fluctuaciones de la intensidad de luz dispersada con el tiempo. La velocidad de tales fluctuaciones está relacionada con el comportamiento con el que las partículas se difunden y este, a su vez, con la forma y el tamaño de las mismas.

La estimación del tamaño de partículas por esta técnica se ve afectada por la presencia de partículas por debajo del micrómetro y la técnica solo ofrece buenos resultados cuando la concentración de sólidos es baja (Heinrich and Ulrich, 2012). Estas limitaciones hacen que la dispersión dinámica de luz no sea una técnica adecuada para la monitorización de suspensiones de microalgas.

Difracción láser

La difracción láser es una técnica que permite estimar las distribuciones del tamaño de partícula a partir de la variación angular en la intensidad de la luz dispersada cuando un haz de láser atraviesa una suspensión de partículas.

Las partículas grandes dispersan la luz en ángulos pequeños respecto a la trayectoria del haz incidente, mientras que las pequeñas dispersan la luz en ángulos mayores.

Los datos de intensidad angular de dispersión se someten a un proceso de deconvolución con la distribución de tamaños de partículas a través de las teorías de dispersión de la luz de Mie y Fraunhofer. La difracción láser como técnica de monitorización de suspensiones de microalgas presenta limitaciones importantes. Por un lado, el patrón de dispersión asociado a una suspensión depende no solo del tamaño de las partículas, sino también de la morfología de las mismas (Ma et al., 2000), lo que puede inducir a errores sistemáticos en la caracterización de la suspensión si la morfología de las partículas no se corresponde a la supuesta para la extracción de la distribución de tamaños. En el caso de microalgas floculadas, la morfología de los flóculos es variable y, como se describe más adelante, está asociada a las condiciones en las que se produce la agregación de células. Por ende, en suspensiones de

microalgas con concentraciones de biomasa relativamente altas, el efecto de solape entre partículas impondría grandes limitaciones a la exactitud de la medida de dispersión. Estas limitaciones desaconsejan el empleo de la difracción láser como técnica de monitorización de suspensiones de biomasa de microalgas.

Medida por reflexión de haz de láser enfocado (FBRM, *Focused Beam Reflectance Measurement*)

La medida por reflexión de haz enfocado permite reconstruir la distribución de tamaño de suspensiones de partículas a través de la retrodispersión de un haz de láser. El principio básico de esta técnica es hacer incidir un haz láser rotatorio sobre la suspensión a caracterizar y registrar tanto el tiempo durante el cual el láser retrodispersado se recibe de vuelta en el detector de la propia sonda como el número de eventos de retrodispersión detectados. Dado que la velocidad de rotación del haz es conocida, con los datos de tiempo y número de eventos de intersección entre el haz y las partículas, el equipo es capaz de construir una distribución de las distancias, o tamaños de cuerda, recorridas por el láser sobre la superficie de las partículas.

La medida por reflexión de haz de láser enfocado presenta una serie de ventajas que la hacen especialmente adecuada para la caracterización de suspensiones de biomasa microalgas. Por un lado, las mediciones realizadas con esta técnica no se ven afectadas por el factor de forma de las partículas en la misma medida que la difracción láser y detectan partículas de más de 1 μm hasta varios milímetros. Además, las medidas por FBRM son fiables a concentraciones que varían de 0.1 a 40% vol. (Mersmann, 2001; Preikschat and Preikschat, 1991) .

La desventaja más notable que presenta esta técnica es la tendencia de esta medida a subestimar el tamaño de las partículas (Li and Wilkinson, 2005), especialmente si estas presentan formas muy elongadas, dado que la probabilidad de intersección del haz de láser con las partículas es más probable en segmentos de estrechos de superficie que a lo largo de longitudes próximas al eje mayor de las mismas.

Durante las últimas décadas el grupo de investigación ha realizado numerosas investigaciones sobre procesos de agregación de partículas desarrollando metodologías y modelos basados en las medidas de la técnica FBRM que han permitido estudiar en profundidad los procesos de floculación (Alonso Zorita, 2014; Bianco et al., 2002; Fuente et al., 2003). La técnica se ha aplicado a diferentes entornos industriales como son la industria papelera (Antunes et al., 2015; Nicu et al., 2012; Zakrajšek et al., 2009), de cemento (Ferron et al., 2013), el tratamiento de aguas (Miranda et al., 2008; Saarimaa et al., 2006), la producción de fibrocemento (Jarabo et al., 2010; Jarabo et al., 2012; Negro et al., 2005; Negro et al., 2006), producción de nanocelulosas (Raj et al., 2016; Raj et al., 2017) e incluso en el cosechado de microalgas (Uduman et al., 2010b). El potencial de la técnica FBRM para la monitorización de bioprocesos ha sido reconocido por Höpfner et al., 2010 y Kara et al., 2011. Si bien no existen muchas referencias en esta línea, cabe destacar que la técnica se ha aplicado en algún caso al estudio de suspensiones de cultivos celulares (Jeffers et al., 2003), llegando a abordarse la estimación de la concentración de suspensiones de cultivos celulares de plantas con FBRM (McDonald et al., 2001) correlacionando la suma de todos los tamaños de cuerda ponderados al cubo con la concentración de biomasa. En este trabajo se indica, precisamente, que los

problemas prácticos encontrados en la estimación de la biomasa por causa de la variabilidad de tamaño y forma de los agregados se pueden solventar aplicando técnicas de regresión no lineales entre la concentración y la distribución de tamaños de cuerda.

En esta tesis se pretende utilizar la técnica FBRM para el estudio de la agregación de microalgas como herramienta de hardware básica para el desarrollo de tres nuevos métodos de caracterización de suspensiones de microalgas que, de acuerdo a nuestro conocimiento no se han descrito previamente en la literatura, siendo estas investigaciones pioneras en este campo.

El FBRM se emplea en esta tesis como herramienta de hardware básica para el desarrollo de métodos de caracterización de suspensiones de flóculos de biomasa de microalgas y también en la monitorización de los ensayos de floculación de suspensiones de microalgas con celulosa nanocristalina *hairy* cationizada.

Microscopía óptica convencional

El microscopio óptico compuesto consta de dos sistemas de lentes y un sistema de iluminación para conseguir la ampliación de la muestra considerada. El acoplamiento de una cámara digital al microscopio permite la adquisición de imágenes que se pueden analizar posteriormente para la caracterización de las suspensiones. Las ventajas principales de esta técnica es su facilidad de uso y coste relativamente bajo. Los principales inconvenientes de la microscopía convencional son el tiempo requerido para el procesamiento de las imágenes previo a la obtención de los resultados y la relativamente baja profundidad de campo que se obtiene en las imágenes.

Esta técnica se emplea en la presente investigación para la caracterización geométrica de agregados de microalgas.

Microscopía confocal láser de barrido

Esta técnica de microscopía se basa en el empleo de fluorescencia para la adquisición de imágenes. En un microscopio confocal la iluminación de la muestra se concentra en un único punto del plano óptico (Webb, 1996) y la imagen se adquiere a través de un orificio estenopéico frente al sistema de detección de luz, en la actualidad, un sensor digital. De esta manera la luz que procede de zonas de la muestra fuera del plano focal no se registra, con lo que se obtiene una imagen de gran calidad y resolución.

Como el sistema de iluminación se concentra en un punto en cada momento, la muestra ha de someterse a un escaneo a lo largo de toda el área considerada del plano focal (barrido) (Paddock, 2000). Un recorrido por varios planos focales a distinta altura en la muestra permite la reconstrucción tridimensional de la misma tras el procesado por medio de herramientas de software especializadas.

Esta técnica permite la adquisición y el procesado de imágenes de biomasa de microalgas con gran calidad y sin necesidad de añadir agentes de tinción al objeto de promover la fluorescencia de la muestra, por ser la clorofila autofluorescente (Millach et al., 2017).

Los inconvenientes principales de esta técnica son el tiempo de adquisición y procesamiento de las imágenes. Por otro lado, si bien a la hora de caracterizar la morfología de las partículas

de biomasa de microalgas la reconstrucción tridimensional de objetos que posibilita esta técnica es muy atractiva, cuando las partículas superan un cierto tamaño, no se pueden resolver los detalles del interior de las mismas, resultando la reconstrucción impracticable.

Esta técnica se emplea en la presente investigación para la caracterización geométrica de agregados de microalgas.

1.3.3.2. Técnicas de estimación de la dimensión fractal

A continuación se describen someramente las principales técnicas de estimación de la dimensión fractal de agregados.

Dispersión de luz

Las técnicas de estimación por dispersión de luz se basan en la interpretación de la radiación secundaria que se produce cuando las partículas son irradiadas con una fuente de luz primaria. La medida de la radiación dispersada en función del ángulo de dispersión proporciona información sobre la forma en que se distribuye la materia en las partículas, en función de la escala longitudinal. Para llevar a cabo la caracterización estructural de las partículas se han empleado diversos tipos de radiación, si bien la luz visible es la fuente cuyo uso es más sencillo y barato. Para obtener la dimensión fractal de los patrones de dispersión de las partículas se emplea la teoría de Rayleigh-Gans-Debye (Liao et al., 2005). El principio básico de esta teoría es modelar el objeto causante de la dispersión de la luz como una colección de objetos dispersores de Rayleigh que no interactúan entre sí. La onda dispersada total será pues la suma de vectores de las ondas dispersadas por cada uno de los objetos componentes.

Los principales inconvenientes para la aplicación industrial de esta técnica para la caracterización de suspensiones de microalgas son el efecto de solape entre los agregados en las suspensiones de microalgas y el tamaño de los mismos, que impide aplicar con exactitud de la teoría Rayleigh-Gans-Debye (Liao et al., 2005).

Ensayos de sedimentación

La dimensión fractal de los agregados se puede estimar a través de la interpretación del comportamiento de los mismos en procesos de sedimentación (Johnson et al., 1996) Para llevar a cabo esta estimación, se determinan experimentalmente la velocidad de sedimentación y el tamaño del agregado y estos parámetros se relacionan con la dimensión fractal a través de un modelo de sedimentación que ha de asumirse previamente (Tang et al., 2002).

La aplicación de esta técnica a escala industrial para la caracterización morfológica de suspensiones de biomasa de microalgas está limitada por la lentitud y falta de portabilidad de los ensayos a realizar, así como por la incertidumbre que conlleva la suposición de un modelo de sedimentación.

Análisis de imagen

Con independencia del método de adquisición de imágenes que se emplee, la estimación de la dimensión fractal de agregados se puede llevar a cabo mediante un análisis de la geometría a varias escalas de los agregados a través de dichas imágenes. El proceso de análisis

normalmente conlleva la detección de los agregados presentes en la imagen y el pretratamiento de las mismas para obtener una versión binarizada de las mismas. Una vez seleccionados los agregados binarizados se realiza un análisis numérico de los mismos. El método más popular de análisis es el conteo de cajas.

El método de conteo de cajas se basa en registrar el número de cajas de tamaño característico ε requerido para cubrir el área completa de la proyección del objeto fractal y repetir el proceso a distintos tamaños característicos. La dimensión fractal proyectada del objeto se obtendría de la expresión siguiente (ecuación 1.5)

$$D_2 = \lim_{\varepsilon \rightarrow 0} \left(\frac{\log(N(\varepsilon))}{\log(1/\varepsilon)} \right) \quad (1.5)$$

Donde $N(\varepsilon)$ es el número de cajas requeridas con el tamaño característico ε .

En términos prácticos, la dimensión fractal proyectada se obtendría de la pendiente de la recta de regresión resultante de representar el logaritmo del número de cajas a distintas escalas de tamaño frente al logaritmo del inverso del tamaño característico, como se muestra en la ecuación 1.6.

$$\log(N(\varepsilon)) = D_2 \log(1/\varepsilon) \quad (1.6)$$

En la literatura se puede encontrar un gran número de modificaciones del método de conteo de cajas pero todas se basan en el principio anteriormente expuesto (Bisoi and Mishra, 2001; So et al., 2017).

A pesar de la sencillez de su aplicación, los métodos de análisis de imágenes proporcionan como resultado la dimensión fractal proyectada, D_2 , que si bien es un parámetro que da indicación sobre la estructura de las partículas, la información que aporta es limitada, pues no existe una correspondencia unívoca entre la dimensión fractal en tres dimensiones, D_f , y la proyectada (Orhan et al., 2016). A esto hay que añadir que el valor de la D_2 estimada depende de la orientación de la partícula en el momento de la adquisición de la imagen (Wentzel et al., 2003).

Otras técnicas

Cabe señalar en este punto que existen estudios en la bibliografía que abordan la estimación de la dimensión fractal por otras técnicas. Muy interesante, y relacionado con las investigaciones de la presente tesis doctoral, es el trabajo de Kovalsky and Bushell (2005). En este estudio se lleva a cabo la estimación de la dimensión fractal de esferas de látex floculadas correlacionándola con la transformada rápida de Fourier del perfil de intensidad reflejado obtenido a través de una sonda de reflectancia láser modificada. Aunque el método fue inexacto para estimar la geometría de los flóculos, los autores demostraron que la señal de reflectancia del láser sin procesar contenía información estructural de los agregados.

1.3.3.3. Estimación de la concentración y geometría media de suspensiones de microalgas a partir de distribuciones de tamaños de cuerda

Un enfoque que se explota en varias investigaciones enmarcadas en la presente tesis doctoral concierne a la interpretación de los datos de tamaños de cuerda de suspensiones de microalgas, bien al objeto de estimar la concentración de la biomasa o con el propósito de estimar la dimensión fractal media de los agregados de microalgas. La correlación entre los espectros de tamaños de cuerda y los dos parámetros referidos se llevó a cabo mediante la aplicación de técnicas de inteligencia artificial, en concreto de modelos de aprendizaje automático. En la estimación de la concentración de biomasa se emplearon redes neuronales artificiales del tipo perceptrón, bosques aleatorios y máquinas de soporte de vectores, y en la estimación de la dimensión fractal se emplearon modelos de bosques aleatorios.

La estimación de la concentración de biomasa microalgas a través de la distribución de los tamaños de cuerda, por ser rápida e independiente de la turbidez del medio pueden ser de utilidad en la monitorización de cultivos de microalgas cuando existe agregación y cuando el medio presenta una turbidez variable, como puede ser el caso al cultivar microalgas con aguas residuales.

El método de estimación directa de la morfología de los agregados de microalgas que resultan de los procesos de floculación podría aplicarse on-line, con lo que su implementación posibilitaría el establecimiento de sistemas de control de la floculación orientados a garantizar una geometría óptima para los procesos de separación que siguen al de floculación.

A continuación se describen brevemente los modelos de aprendizaje automático empleados en el desarrollo de los sistemas de estimación mencionados.

Redes neuronales artificiales de tipo perceptrón multicapa

El concepto de perceptrón fue introducido por primera vez por Rosenblatt (1958) con la idea de proponer un modelo representativo de la forma en la que la información sensorial se almacena en la memoria de los organismos superiores. El perceptrón se estructura en neuronas artificiales, cuya estructura fue propuesta por primera vez por McCulloch and Pitts (1943). La neurona artificial es una función matemática que recibe una serie de parámetros de entrada, los procesa y devuelven una salida. El perceptrón multicapa, que es el empleado como sistema de aprendizaje automático presenta, como mínimo, tres capas de neuronas, una capa de entrada, una capa oculta y otra de salida. En función de su ubicación en la estructura, las neuronas pueden recibir la señal de entrada directamente de las variables de entrada o de otras neuronas. La neurona procesa la suma de todas las señales de entrada ponderadas y le aplica una función, la función de transferencia, a dicha suma cuyo resultado es la señal de salida, como ejemplifica la ecuación 1.7.

$$y_i = \varphi\left(\sum_{j=0}^m w_{ij}x_j + b_i\right) \quad (1.7)$$

En esta ecuación, y_i es la señal de salida de la neurona i , φ es la función de transferencia. El sumatorio se aplica a los valores de salida, x_j , de las neuronas que se conectan a la neurona i (o de la entrada a la red) multiplicadas por el peso, w_{ij} que conecta la neurona j a la i , o en su caso uno de los elementos de la variable de entrada y b_i la ganancia de la neurona. El proceso de aprendizaje conlleva el ajuste de los pesos de la red de forma que al alimentar la red con

valores de la variable de entrada conocidos se obtenga como resultado los valores de variable de salida conocida con los que se corresponden. Este ajuste se lleva a cabo por retropropagación de errores, de manera que se van actualizando los pesos de las neuronas a partir del error obtenido en la última capa de la red, que es la que devuelve los valores de salida (Riedmiller, 1994). A partir de la red entrenada, la red sería capaz de estimar con un grado de precisión aceptable los valores de salida a partir de valores de entrada no presentados previamente al algoritmo.

Máquinas de vectores de soporte para regresión

Las máquinas de vectores de soporte para regresión (SVR, por su denominación en inglés, *support vector regression*) se basan en la teoría del aprendizaje estadístico (Vapnik, 1995) y fueron propuestas por primera vez por Drucker et al. (1997). Las SVR presentan ciertas características que hacen preferible a las redes neuronales artificiales de tipo perceptrón. Primero, la función objetivo de las SVR es convexa, lo que implica que alcanzar el óptimo global está garantizado, lo que no siempre es así en el caso del perceptrón (Thissen et al., 2004). Por otro lado, dado que el algoritmo SVR intenta minimizar el error generalizado (una combinación del error de entrenamiento y una regularización de la complejidad del modelo) en lugar del error de entrenamiento, los modelos SVR pueden evitar el sobreajuste en gran medida (Basak et al., 2007). Por último, la optimización del modelo, no depende de la dimensionalidad de los datos de muestra (Drucker et al., 1997). El algoritmo SVR tiene como objetivo ajustar una función $y = f(x)$ de manera que el parámetro y se pueda predecir con precisión cuando a la función se le presentan muestras de x que no se le han presentado previamente. Las SVR se basan en realizar un mapeo de los vectores de muestra (en el caso presente, los vectores de distribución de tamaños de cuerda) en un espacio dimensional superior apropiado en el que es posible realizar una regresión lineal con los datos transformados. El modelo producido por SVR solo depende de un subconjunto de los datos de entrenamiento (que contienen los vectores de soporte) porque la función de coste derivada de él ignora cualquier muestra de entrenamiento que esté cerca (dentro de un umbral determinado ϵ) a la predicción del modelo. Los hiperparámetros de SVR sintonizados para buscar el modelo óptimo o casi óptimo donde funciona el núcleo, el equilibrio entre el error de entrenamiento y la complejidad del modelo (C), el ancho del tubo insensible al error (ϵ) y un parámetro, gamma, que es una constante asociada a la función del *kernel* empleado para realizar el mapeo a espacios dimensionales mayores.

Bosques aleatorios

Los bosques aleatorios de regresión (RFR por su nomenclatura en inglés, *random forest regression*) son meta-heurísticos de aprendizaje automático introducidos por primera vez por Breiman (2001). El método se basa en la construcción y el entrenamiento de múltiples árboles de regresión creados al azar. El valor estimado a partir de muestras no presentadas anteriormente al modelo es el resultado de promediar las predicciones realizadas para las mismas por todos los árboles del conjunto. Los árboles de regresión son estructuras construidas a través de un algoritmo que hace una partición recursiva los datos de entrenamiento en grupos o nodos más pequeños, mutuamente excluyentes. El algoritmo recorre cada división posible y selecciona la partición que divide el conjunto de datos en dos grupos de tal forma que se minimizan las desviaciones (residuos) de la media en los

subconjuntos separados. El procedimiento de partición se aplica, seguidamente, a cada nueva rama y continúa hasta que cada nodo alcance un tamaño de nodo mínimo especificado por el usuario, convirtiéndose en un nodo terminal u hoja. La combinación de árboles aleatorios que implementa el algoritmo de bosque aleatorio tiene como objetivo evitar el ajuste excesivo de los datos. En la presente tesis, los hiperparámetros del modelo que se optimizaron para encontrar el modelo de regresión óptimo o cercano al óptimo fueron el número de árboles, la profundidad máxima de los árboles, el número mínimo de puntos de muestra para dividir un nodo y el máximo número de variables explicativas consideradas en la búsqueda de la mejor división en cada árbol.

2

OBJETIVOS

La producción masiva de biomasa de microalgas, por la diversidad de usos posibles que esta presenta, podría contribuir efectivamente a asegurar la provisión alimentaria, a aumentar la sostenibilidad de la producción energética y a disminuir el impacto ambiental vinculado a la producción de alimentos y energía. No obstante, por el momento, los costes de operación asociados a la producción de microalgas constituyen una barrera para la producción masiva de estos microorganismos, siendo las dos etapas con margen de mejora en términos económicos las de cultivo y concentración de la biomasa. Al objeto de abaratar la etapa de cultivo se están realizando importantes esfuerzos de investigación para conseguir el reciclado de los medios de crecimiento a través de tratamientos internos y el empleo de aguas residuales tratadas como medio de crecimiento en virtud a su riqueza en nutrientes. Por otro lado, la tendencia actual para conseguir abaratar la etapa de concentración se orienta hacia disminuir los caudales de biomasa a concentrar mediante una etapa previa de pre-concentración, normalmente una operación de coagulación/floculación por su bajo coste.

El objetivo de la presente tesis doctoral es generar nuevos conocimientos y aportar contribuciones técnicas concernientes a la monitorización de suspensiones de microalgas que puedan contribuir, en la medida de lo posible, a facilitar la transición del proceso de producción de esta biomasa desde el laboratorio o la planta piloto hasta la escala industrial incidiendo en la etapa de cultivo, especialmente con medios alternativos, y en la de pre-concentración por floculación. En el desarrollo de las soluciones de monitorización que se han concebido se ha procurado integrar las necesidades de la industria en lo relativo a la inmediatez de las medidas, tratando, cuando ha sido posible, que las determinaciones puedan realizarse en tiempo real. Para ello, ha sido fundamental el empleo de una sonda de medición de la reflexión de un haz de láser enfocado que permite la adquisición en tiempo real de datos de tamaño de cuerda de partículas en suspensión. Asimismo, dados los avances que ha experimentado la metrología con el crecimiento de los sistemas de inteligencia artificial de

aprendizaje automático, se ha decidido incluir técnicas de esta área de la ingeniería de software en los desarrollos considerados.

El objetivo planteado se divide en varios objetivos específicos. En primer lugar, se pretende conseguir el desarrollo de modelos de correlación que permitan estimar la concentración de biomasa de microalgas en suspensiones con distintos estados de agregación empleando datos de la distribución de tamaños de dichas suspensiones. Este aspecto se concreta en los siguientes tres sub-objetivos:

- Creación de un modelo de correlación de los tamaños de cuerda de suspensiones de agregados de la microalga *Chlamydomonas reinhardtii* con la concentración de biomasa seca de dicha suspensión basado en redes neuronales de tipo perceptrón.
- Creación de un modelo de correlación de los tamaños de cuerda de suspensiones de agregados de la microalga *Chlorella sorokiniana* cultivada en agua residual tratada con la concentración de biomasa seca de dicha suspensión basado en bosques aleatorios.
- Creación de un modelo de correlación de los tamaños de cuerda de suspensiones de agregados de la microalga *Chlorella sorokiniana* cultivada en agua residual tratada con la concentración de biomasa seca de dicha suspensión basado en máquinas de vectores de soporte.

Los resultados de la consecución de los tres objetivos enumerados servirían para llevar a cabo la monitorización de la concentración de cultivos de microalgas muy concentrados o aquellos en los que el medio presenta turbidez y esta puede cambiar a lo largo del proceso.

En segundo lugar, en esta tesis doctoral se pretende desarrollar métodos que permitan caracterizar la forma de los agregados de microalgas producidos por floculación a través de la estimación de la dimensión fractal media de los flóculos de biomasa en suspensión. Este objetivo específico, se divide en los siguientes dos sub-objetivos:

- Creación de sistemas de determinación de la dimensión fractal media de las suspensiones de agregados de microalgas por comparación entre las dimensiones fractales proyectadas de agregados reales procedentes de imágenes microscópicas y las de agregados simulados por ordenador para emular a los reales.
- Creación de sistema de determinación de la dimensión fractal media de las suspensiones de agregados de microalgas mediante la interpretación de la distribución de los tamaños de cuerda de las mismas a través de un modelo de regresión por bosques aleatorios entrenado con datos virtuales.

Como objetivo específico secundario se estudia la efectividad de un nuevo tipo de agente floculante, la celulosa nanocrystalina hairy cationizada (CNC-HC) en la agregación de biomasa de *Chlorella sorokiniana*.

MATERIALES Y MÉTODOS EXPERIMENTALES

3.1. CEPAS DE MICROALGAS Y SU CULTIVO

En la presente investigación se trabajó con dos cepas de microalgas, en concreto una de *Chlamydomonas reinhardtii* (referencia CCAP¹ 11 / 32B) y otra de *Chlorella sorokiniana* (referencia CCAP 211/8 K).

Con el propósito de obtener una cantidad de biomasa cultivada para proceder al escalado de cultivo, ambas cepas se sembraron en matraces de 250 mL previamente autoclavados con 100 mL de medio tris-acetato-fosfato, o medio TAP (Gorman and Levine, 1965).

Este medio presenta la siguiente composición: Tris(hidroximetil)aminometano (20 mM), ácido acético (17.5 mM), NH₄Cl (7.48 mM), MgSO₄ (406 µM), CaCl₂ (340 µM), K₂HPO₄ (540 µM), KH₂PO₄ (463 µM), H₃BO₃ (184 µM), ZnSO₄ (76.5 µM), MnCl₂ (25.5 µM), FeSO₄ (17.9 µM), CoCl₂ (6.77 µM), (NH₄)₆Mo₇O₂₄ (0.88 µM), CuSO₄ (6.29 µM), y Na₂EDTA (148 µM).

Los matraces de cultivo se sometieron a agitación a 115 rpm en una incubadora termostatzada a 23 ° C y provista de iluminación con una matriz de lámparas LED de color blanco frío (figura 3.1a). La iluminación se aplicó en un ciclo de 12 h.

¹ Culture Collection of Algae and Protozoa SAMS Limited, Scottish Marine institute, OBAN, Argyll, Escocia, Reino Unido



Figura 3.1. a) Incubadora termostatzada con iluminación LED; b) fotobiorreactores de laboratorio empleados.

Cuando los cultivos alcanzan una concentración de biomasa de alrededor de 1 g L^{-1} , el contenido de los matraces se transfiere a los fotobiorreactores de metacrilato de 5.5 L (figura 3.1b) operados a temperatura entre 23 y 25°C con aireación (2 L min^{-1}) y pH 7.5. El pH se controla a través del suministro automático de CO_2 (0.2 L min^{-1}). La iluminación de los fotobiorreactores se proporcionó por medio de cuatro lámparas fluorescentes de color blanco frío en un ciclo de 12 h. Tanto la iluminación como el suministro automático de CO_2 se controlan a través de un ordenador conectado a una tarjeta controladora USB LabJack U12 con capacidad para la adquisición de datos de las sondas de pH y de envío de señales al armario de control para actuar sobre las lámparas fluorescentes y la electroválvula de apertura del canal de CO_2 . En la figura 3.2 se ofrece una representación esquemática del dispositivo experimental empleado para el cultivo de microalgas en los mencionados fotobiorreactores.

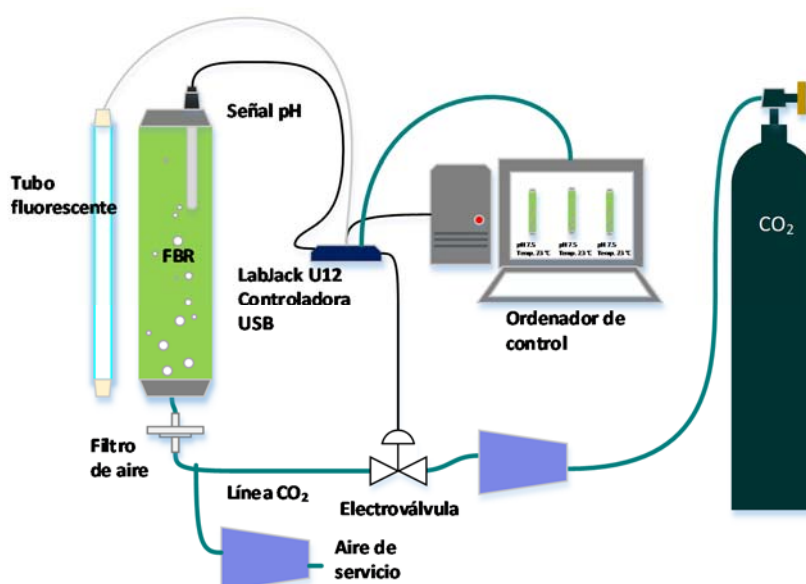


Figura 3.2. Dispositivo experimental para el cultivo de microalgas en fotobiorreactores.

En el cultivo de las cepas mencionadas en fotobiorreactores se emplearon dos medios diferentes. La cepa de *C. reinhardtii* se cultivó solamente con medio TAP ozonizado durante 15 minutos. La cepa de *C. sorokiniana* se cultivó para la realización de dos bloques experimentales. En el primero fue cultivada con agua residual industrial filtrada por 0.22 μm (DQO = 219 mg L⁻¹, turbidez = 387 NTU) procedente de un reactor de digestión anaeróbica y en el segundo con medio TAP ozonizado durante 15 minutos.

3.2. FLOCULANTES

3.2.1. Quitosano

Para conseguir la agregación de las células de microalgas se emplearon dos tipos distintos de quitosano. Se utilizó quitosano de conchas de cangrejo (Sigma-aldrich 48165, Número CAS 9012-76-4) en los ensayos conducentes al desarrollo de modelos de estimación de la biomasa seca de microalgas y quitosano de peso molecular medio (Sigma-Aldrich 448877, número CAS 9012-76-4) en los ensayos conducentes al desarrollo de modelos para la estimación de la dimensión fractal de los agregados de microalgas. El empleo de dos quitosanos diferentes hay que atribuirlo a la descatalogación del primero de entre los productos ofrecidos por el distribuidor. El segundo tipo de quitosano presenta, según el distribuidor, unas características muy semejantes al primero.

En los dos casos, el floculante se preparó de la misma manera. La solución de floculante se preparó disolviendo el quitosano en una solución de ácido acético glacial al 1% vol. La solución se agitó mecánicamente a 400 rpm durante 1 hora y se dejó reposar durante 24 horas antes de su utilización.

3.2.2. Celulosa nanocristalina *hairy* cationizada (CNC-HC)

La materia prima utilizada para la preparación de la nanocelulosa cristalina *hairy* cationizada (CNC-HC) fue una pasta kraft de madera blanda suministrada por Domtar, Canadá. Los productos químicos utilizados para las diferentes reacciones, (meta) peryodato de sodio, cloruro de GT [(2-hidrazinil-2-oxoetil) –trimetilazinio, GT], etilenglicol, hidrocloreuro de hidroxilamina, AgNO₃ y ácido clorhídrico (0.1 M) e hidróxido de sodio (0.1 M) fueron suministrados por Sigma Aldrich (grado analítico de reactivo) y se usaron sin modificaciones adicionales. El propanol y el cloruro de sodio se compraron de ACP Chemicals Inc.

La CNC-HC se preparó a través de un tratamiento de tres pasos basado en el procedimiento propuesto por (Yang and van de Ven, 2016) que consiste en una oxidación de la pulpa con NaIO₄ para producir celulosa modificada con di-aldehído, cationización de la misma para obtener celulosa modificada con di-aldehído catiónica y tratamiento térmico para nanofibrillar este último compuesto y producir CNC-HC.

3.3. DETERMINACIÓN DE LA BIOMASA SECA

Se empleó un volumen de medio de cultivo de 20 mL para cada determinación gravimétrica. Estos volúmenes se tomaron a su vez de un volumen de 2 L extraído del correspondiente

fotobiorreactor que se mantuvo en agitación a 200 rpm durante el proceso al objeto de garantizar la homogeneidad del muestreo.

La determinación de la biomasa seca se llevó a cabo por triplicado siguiendo el método descrito en Beckmann et al. (2009). Las muestra de 20 ml se centrifugaron en una centrífuga Hettich Universal 320 a 6800 RCF durante 15 min y se lavaron dos veces con agua ultrapura. Los *pellets* de biomasa obtenidos se secaron a 101.5 ° C durante 3 h en placas de aluminio previamente taradas. Después del secado, los platos se colocaron en un desecador durante 45 minutos y luego se volvieron a pesar. La concentración de biomasa seca de microalgas se calculó en g L⁻¹ de medio de cultivo.

3.4. ENSAYOS DE FLOCULACIÓN

La floculación se llevó a cabo con muestras de cultivo de 200 mL aplicando una agitación y dosis de quitosano diferentes según el bloque experimental a que perteneciera el ensayo. A continuación se describe el procedimiento seguido en cada bloque.

- Desarrollo de modelos para la estimación de la concentración de biomasa seca: Se empleó agitación a 200 rpm y distintas dosis de quitosano al 1% para conseguir la floculación parcial de la muestra.
- Desarrollo de modelos para la estimación de la concentración de biomasa seca: La agitación se logró por medio de un impulsor de paso de cuatro palas de 4,95 cm de diámetro ($N_p = 1,27$). Se consideraron tres velocidades de agitación, concretamente 200, 350 y 500 rpm, que correspondían a los números de Reynolds $8,2 \times 10^3$, $1,4 \times 10^4$ and $2,0 \times 10^4$ y gradientes de velocidad (G) 263, 611 y 1043 s⁻¹, respectivamente. En cada caso, se añadió una dosis de 5 ppm de solución de quitosano a la muestra de microalgas agitada. En los ensayos que servirían para la toma de imágenes microscópicas de agregados, tras los dos minutos de agitación, se añadieron 0,5 ml de undispersante comercial (Nopco ESA 120) a la muestra para evitar la agregación de los flóculos en el periodo comprendido entre la toma de muestra y la adquisición de imágenes.

3.5. OBTENCIÓN DE DATOS DE DISTRIBUCIONES DE TAMAÑOS DE CUERDA DE LAS SUSPENSIONES

Se usaron dos sondas de adquisición de DTC, ambas del fabricante Mettler-Toledo, la FBRM M500L (figura 3.3a) y la Particle Track G400 (G400) (figura 3.3b).

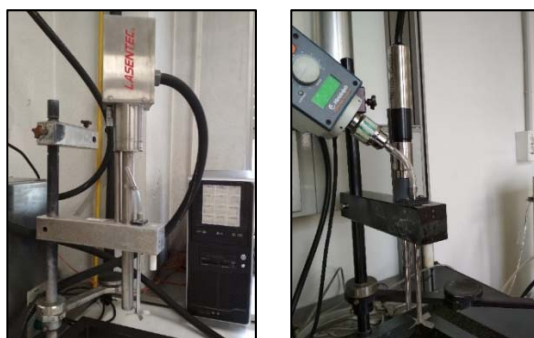


Figura 3.3. Equipos de análisis del tamaño de cuerda, a) sonda FBRM M500L y b) PT G400.

Ambas sondas se basan en los mismos principios de funcionamiento, como se recoge en la figura 3.4.

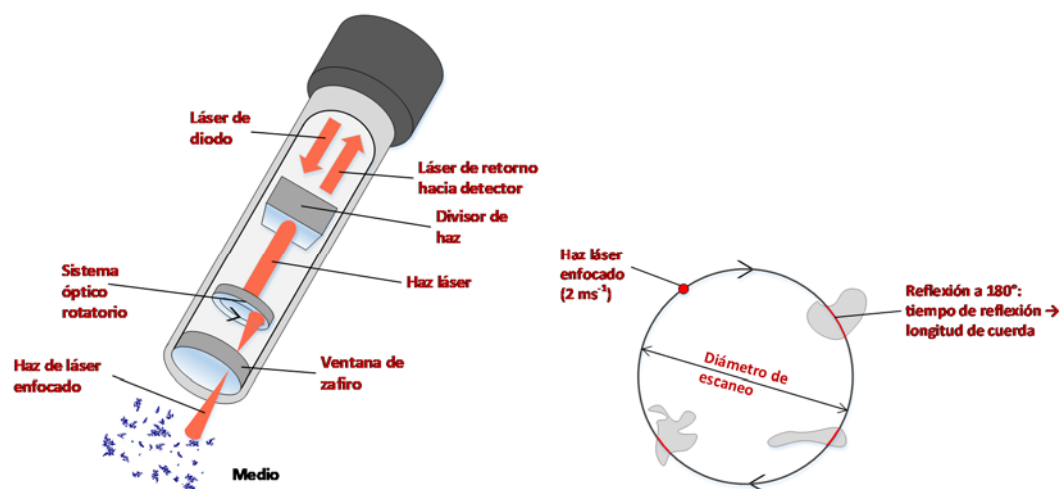


Figura 3.4. Esquema de funcionamiento de la sonda FBRM.

La medida se basa en la proyección de haz de láser enfocado que gira a velocidad constante sobre el seno de la muestra líquida. Dado que la velocidad de giro del haz de láser es conocida, la sonda calcula la longitud de cuerda de las partículas a partir del tiempo durante el cual la partícula refleja el haz incidente. La elevada velocidad de giro del haz permite recoger un gran número de eventos de reflexión asociados a longitudes de cuerda. Estos datos, organizados por frecuencia y en intervalos de tamaño, constituyen la distribución de tamaños de cuerda de la suspensión considerada.

La diferencia más notable de las dos sondas empleadas concierne al intervalo de medida. La sonda M500L es capaz de detectar agregados desde 1 hasta aproximadamente 1000 μm (De Clercq et al., 2004), mientras que la G400 detecta tamaños de cuerda desde 0,5 hasta 4000 μm (Zhou et al., 2014). En diámetro de escaneo en ambos equipos es también distinto, siendo de 8,5mm en la M500L (frecuencia de escaneo 4500 Hz) y 5 mm en el caso de la G400 (frecuencia de escaneo 7200 rpm). Por otro lado, de acuerdo con el proveedor, el haz de láser emitido por la G400 es unas diez veces más potente que en del equipo más antiguo y el sistema de detección entre 15 y 20 veces más sensible. El software de ambos dispositivos permite organizar los resultados de frecuencia de tamaños de cuerda en función de intervalos logarítmicos de tamaño.

Se emplearon dos tipos de distribuciones de tamaño en esta tesis doctoral, cada uno de ellos asociado a las sondas indicadas:

- Ensayos con *C. reinhardtii*: 90 intervalos de longitud organizados logarítmicamente desde 1 hasta 1000 μm .
- Ensayos con *C. sorokiniana*: 200 intervalos de longitud organizados logarítmicamente desde 1 a 4000 μm .

En todos los casos se tomaron muestras de 200 mL y la sonda FBRM se colocó siempre perpendicular a la superficie de la suspensión. La agitación se consiguió por medio de un agitador de 4 palas con inclinación de 45° y diámetro 4,95 cm.

En el caso de los ensayos de estimación de la concentración de biomasa de microalgas, se aplicó siempre una velocidad de agitación de 200 rpm. Para reproducir situaciones de correspondencia no unívoca entre la concentración de biomasa (CB) y distribución de tamaños de cuerda (DTC), las muestras de microalgas frescas se floclaron parcialmente con diferentes dosis de quitosano al 1%, concretamente 10, 20, 30 ppm. De esta forma, generamos diferentes estados de agregación y sus correspondientes datos de tamaños de cuerda correspondientes a una misma concentración de biomasa.

3.6. ADQUISICIÓN DE IMÁGENES POR MICROSCOPÍA ÓPTICA CONVENCIONAL

Las imágenes microscópicas de las muestras de cultivo de microalgas se tomaron mediante un microscopio Axio Lab.A1 (Carl Zeiss) con cámara digital integrada AxioVisio, empleando portamuestras con pocillo (figura 3.6).

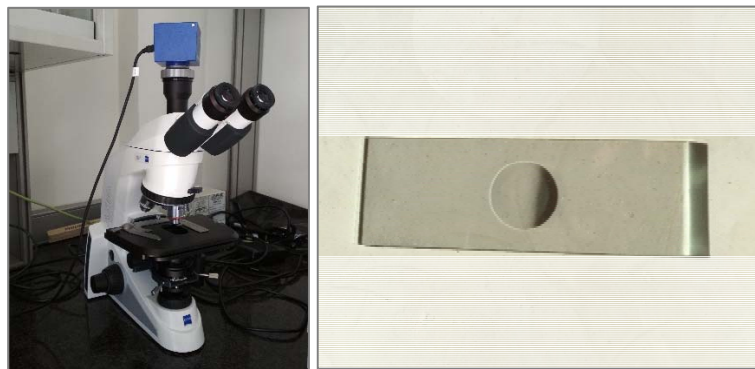


Figura 3.5. Microscopio Axio Lab.A1 y portamuestras con pocillo.

3.7. ADQUISICIÓN DE IMÁGENES POR MICROSCOPÍA CONFOCAL LÁSER DE BARRIDO

Las imágenes microscópicas agregadas se tomaron con un microscopio láser confocal Olympus FV1200, usando una excitación láser de 405 nm y recibiendo la emisión desde 594 nm. No se usó ninguna preparación de muestra especial para la adquisición de imágenes: se colocó una gota de la muestra de microalgas en un portaobjetos por medio de una pipeta Pasteur de punta ancha para evitar dañar los agregados. Los flóculos se dejaron sedimentar en la parte inferior del portaobjetos antes de la adquisición de la imagen. La sedimentación se traduce en un aumento de las posibilidades de que los flóculos presenten el área de superficie máxima proyectada en el plano horizontal.

4

MÉTODOS COMPUTACIONALES

4.1. MODELOS DE APRENDIZAJE AUTOMÁTICO APLICADOS A LA REGRESIÓN

Se emplearon tres tipos de modelos de regresión basados en técnicas de aprendizaje automático, en concreto, el perceptrón multicapa, el bosque aleatorio y las máquinas de vectores de soporte. Se empleó el perceptrón por ser un método cuya aplicación en la ciencia e ingeniería cuenta con una larga trayectoria. Se decidió aplicar los otros dos modelos en virtud a su capacidad para realizar tareas de clasificación y regresión con precisión en las que el número de variables predictoras es alto en comparación con las muestras disponibles en el conjunto de datos (Cosgun et al., 2011; Zheng et al., 2009).

La arquitectura de cada modelo se optimizó mediante la selección de los hiperparámetros correspondientes más adecuados en cada caso. El proceso de optimización de la red neuronal artificial tipo perceptrón se describe más adelante en la sección 4.2.3 y la aplicada a los modelos de regresión por bosques aleatorios y máquinas de vectores de soporte en la sección 4.2.4. En las tablas 4.1 a 4.3 se indican los hiperparámetros objeto de optimización en cada tipo de modelo para la red neuronal de tipo perceptrón, la regresión por máquina de vectores de soporte y la regresión de bosques aleatorios respectivamente.

Tabla 4.1. Hiperparámetros de la red neuronal de tipo perceptrón.

Hiperparámetro	Valores considerados
Número de capas ocultas	1 o 2
Número de neuronas en la primera capa oculta	3 a 18
Número de neuronas en la segunda capa oculta (cuando procede)	3 a 18
Sesgo en la primera capa oculta	Sí / no
Sesgo en la segunda capa oculta	Sí / no
Sesgo en la capa de salida	Sí / no
Valor inicial de los pesos desde el entrada a la primera capa oculta	1
	0.1
	0.01
	0.001
Valor inicial de los sesgos en las capas ocultas	1
	0.1
	0.01
	0.001
Intervalo de valores iniciales de los pesos que van desde la primera capa oculta hasta la segunda capa oculta o salida	1
	0.1
	0.01
	0.001
Función de transferencia de la primera capa oculta	Logística
	Tangente hiperbólica
	Sigmoidal
Función de transferencia de la segunda capa oculta	Logística
	Tangente hiperbólica
	Sigmoidal
Función de transferencia de la capa de salida	Logística
	Tangente hiperbólica
	Sigmoidal
Proporción de los errores cuadráticos medios y la media pesos cuadrados (regularización)	1
	0.5
	0.1
	0.01

Tabla 4.2. Hiperparámetros de la regresión por máquina de vectores de soporte.

Hiperparámetro	Valores considerados
Tipo de función kernel	De base radial, sigmoidal, lineal
C	0.01 – 10,000
ϵ	0.01 – 10
Gamma	0.00001 – 3.8

Tabla 4.3. Hiperparámetros de la regresión por bosques aleatorios.

Hiperparámetro	Valores considerados
Número de árboles	10 – 1,000
Máxima profundidad	10 – 70
Mínimos número de muestras permitido	2 – 9
Máximo número de variables para la partición	1 – 200

Los modelos considerados para la regresión por máquina de vectores de soporte y la regresión por bosques aleatorios se construyeron empleando las librerías de aprendizaje automático Scikit-Learn versión 0.17 (Pedregosa et al., 2011) disponibles para Python 2.7.

4.2. ALGORITMO DE OPTIMIZACIÓN DE LA ARQUITECTURA DE LOS MODELOS DE REGRESIÓN

Para llevar a cabo la selección de los valores óptimos para los hiperparámetros de cada modelo se emplearon dos versiones de un algoritmo de optimización de recocido simulado (*simulated annealing*). La primera versión, más sencilla, se aplicó solamente en la optimización de las redes neuronal artificial. La segunda se utilizó en el resto de casos. A continuación se describen brevemente ambos algoritmos.

4.2.1. Algoritmo de recocido simulado

El algoritmo de recocido simulado es un método de estimación del mínimo global de una función en espacios de búsqueda muy amplios. El algoritmo es una modificación del algoritmo de Metropolis–Hastings (Chib and Greenberg, 1995) para obtener muestras aleatorias de una distribución. El algoritmo se basa en la analogía con el proceso de recocido que se da la industria metalúrgica, a través del cuál se calientan las piezas de metal a altas temperaturas para producir una eliminación de las tensiones internas del material que se asocian a una disminución rápida de la energía libre de Gibbs que se ha acumulado cuando el metal se trabaja en frío. Tras el calentamiento se mantiene la temperatura y posteriormente se deja que la pieza se enfríe lentamente (Bohachevsky et al., 1986).

El algoritmo de recocido simulado consiste en ir generando modelos aleatoriamente y evaluando la desviación de la estimación que producen respecto al valor real. El rasgo distintivo de este tipo de metaheurístico estriba en la forma en que el algoritmo acepta o rechaza los modelos generados, puesto que si bien siempre acepta aquel modelo que es más exacto que el anterior, también puede aceptar modelos cuya exactitud es peor que el precedente siempre que se cumpla una cierta premisa.

Para comprender mejor el procedimiento de aceptación o rechazo de modelos, cabe indicar que en el algoritmo de recocido simulado, el error del modelo de regresión, es decir la mencionada desviación entre el valor estimado y el valor real, constituye el análogo de la energía libre de Gibbs del metal real. En un sistema real, la probabilidad de que se dé una transición de un estado de energía a otro depende de la diferencia de energía entre los dos estado y de la temperatura del sistema a través de la siguiente ecuación (Selim and Alsultan, 1991).

$$P(\text{cambio}) = e^{-\frac{\Delta E}{kT}} \quad (4.1)$$

En la ecuación 4.1, ΔE representa la diferencia de energías entre los estados, T la temperatura y k es la constante de Boltzman. En el caso del metaheurístico, la diferencia de energía tiene como análogo la diferencia del error entre un modelo y el precedente y la temperatura tiene

un análogo computacional de valor arbitrario y, al igual que en la realidad, en descenso con cada iteración. En el caso considerado el valor de k se fijó en uno. En el caso de esta investigación se empleó el programa de disminución de la temperatura que describe la siguiente ecuación, siendo α menor que la unidad.

$$T_{i+1} = \alpha T_i \quad (4.2)$$

Dado que la temperatura va disminuyendo en el sistema la probabilidad de aceptar modelos peores que el precedente va decayendo con el transcurso de la ejecución del proceso de optimización de forma que al final del mismo solo se aceptan modelos mejores que el precedente. A pesar de que puede resultar contraintuitiva la aceptación de modelos cuya capacidad de estimación es pobre, lo cierto es que tal propiedad del algoritmo le permite recorrer la región de búsqueda de forma bastante exhaustiva, minimizando la posibilidad de que el sistema caiga en un mínimo local.

4.2.2. Generación aleatoria de modelos y su mutación

Para una mejor comprensión de los procesos de generación y mutación de los modelos es útil recurrir a la terminología que se emplea en el área de la genética. Con tal enfoque, se puede entender el conjunto de los hiperparámetros de un modelo como el genotipo de dicho modelo. Cada hiperparámetro del conjunto cabría pues entenderse como un cromosoma.

La generación aleatoria de un modelo conlleva la selección inicial al azar de su genotipo, es decir del valor de cada uno de los cromosomas que lo componen. La mutación del modelo consiste en aplicar cambios aleatorios a los valores de los hiperparámetros (Dowsland and Thompson, 2012). En la aplicación de un algoritmo de optimización de recocido simulado, la mutación en el genotipo del modelo concierne solamente a un cromosoma.

La forma más sencilla de codificar el valor de un cromosoma, ya sea un número real o un valor categórico, es emplear una cadena de bits. La concatenación de las cadenas de bits constituye pues el genotipo de modelo. Dado el carácter finito de las cadenas de ceros y unos, los valores que estas estructuras pueden codificar están discretizados. Esto implica que previamente a la codificación haya de crearse una lista discreta de valores posibles para el cromosoma considerado. Cada valor de esta lista se hace corresponder mediante una tabla a una cadena de bits de longitud fija.

El número de valores que se puede codificar con este procedimiento es 2^n , siendo n el número de bits, o lo que es lo mismo, la longitud de la cadena que codifica el cromosoma. Aparte de su sencillez, otra ventaja que presenta la codificación mediante bits es la facilidad con que se lleva a cabo la mutación en el modelo. Esta implica la selección aleatoria de una posición dentro del vector de bits que codifica el genotipo completo y la permutación del valor de bit correspondiente. Se plantea el siguiente ejemplo sencillo para la comprensión del proceso.

Codificación de 8 números con tres bits:

000	001	010	011	100	101	110	111
0	1	2	3	4	5	6	7

Consideramos un genotipo con dos cromosomas y los valores 7, 2:

111	010
7	2

Decidimos al azar mutar el segundo cromosoma en la primera posición de la cadena de bits. El resultado sería el siguiente:

111	110
7	6

Existen dos métodos principales de codificación en bits, el sistema binario y el código de Gray. En el sistema binario representa un número mediante combinaciones de ceros y unos en los que la posición de cada dígito está ponderada, es decir que tiene mayor o menor peso en función del emplazamiento que ocupa en la cadena de bits. En el sistema de codificación de Gray, por otro lado, las posiciones de los bits en la cadena no están ponderadas. La ventaja del código de Gray frente al binario natural estriba en que la modificación de bits cercanos en la cadena están asociadas a pequeñas modificaciones en el cardinal que codifican y, por lo tanto se asocian a cambios pequeños en los valores del hiperparámetro (Mathias and Whitley, 1994). En el sistema binario tradicional, un cambio en el bit adyacente pueden comportar un cambio de magnitud mayor en el cardinal que codifican, con lo que los valores de hiperpámetro codificados por bits muy similares pueden ser muy dispares. Si se emplea el código de Gray, las mutaciones en bits cercanos dan lugar a modelos parecidos, lo cuál es deseable para que la búsqueda del mejor modelo tenga un comportamiento homogéneo (Nishijima et al., 2014).

En el presente trabajo, el sistema binario se empleó en la codificación de los modelos de regresión basados en redes neuronales artificiales de tipo perceptrón (*C. reinhardtii*) y el sistema de codificación de Gray se empleó en la codificación del resto de modelos de regresión por aprendizaje automático al objeto de mejorar el proceso de optimización de los hiperparámetros.

4.2.3. Aplicación del algoritmo de recocido simulado sencillo en la optimización de la arquitectura del perceptrón multicapa

La figura 4.1 describe el flujo de información en el algoritmo de búsqueda sencillo por recocido simulado empleado para optimizar la arquitectura de la red neuronal artificial de tipo perceptrón. El algoritmo comienza fijando el valor cero en el contador de iteraciones, la

temperatura inicial y generando un vector aleatorio de bits de longitud prefijada. Este vector se traduce por medio de una función en otro vector que contiene números reales y cadenas de caracteres. Los valores de este vector constituyen los valores iniciales de los hiperparámetros de la red (véase tabla 4.1). Se genera una red con los hiperparámetros iniciales, se entrena con la subpoblación de datos de muestreo de entrenamiento y se valida con los de comprobación. El error resultante de comparar los datos de concentración reales con los estimados a través de los datos de verificación de la red se calcula como error cuadrático medio y se almacena. Al ser el resultado obtenido de la primera red generada, este error se considera el mejor error provisional y la arquitectura de la red, se asume como óptima provisionalmente. Se verifica que el número de iteraciones no se ha alcanzado y se genera una nueva red a partir de una modificación de la anterior. Esta modificación concierne a los hiperparámetros y se induce a través de la mutación aleatoria en uno de los bits que componen el vector que codifica los valores de los hiperparámetros. La mutación consiste en permutar el valor del bit (el bit solo puede tomar el valor cero o uno) cuya posición en el vector se establece al azar. Realizada la modificación aleatoria del vector que codifica la arquitectura del modelo, se procede a su traducción a los valores de los hiperparámetros del nuevo modelo modificado, se construye y se evalúa la desviación de la estimación que produce con respecto al valor real. Se aplica entonces el criterio de aceptación descrito en el apartado anterior. Dependiendo de este criterio, el modelo se acepta como el mejor provisional o no. Si se acepta, se comprueba si la estimación que produce tiene una desviación menor o igual que la consignada y en tal caso se acaba el proceso de optimización. Si el modelo se rechaza, se procede a disminuir la temperatura del sistema y se vuelve al ciclo de mutación aleatoria y evaluación descrito anteriormente.

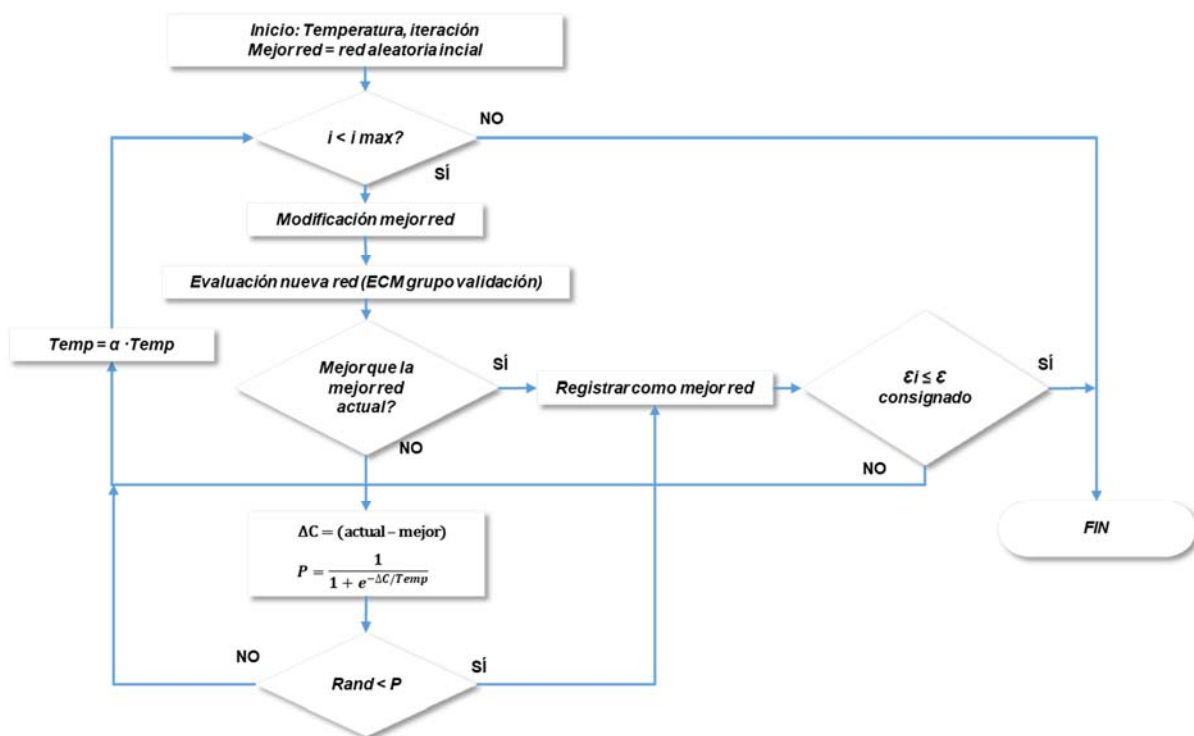


Figura 4.1. Algoritmo de recocido simulado para la optimización de la arquitectura del perceptrón

El algoritmo descrito se programó en Matlab R14 (Mathworsk) empleando las librerías de construcción de redes neuronales artificiales.

4.2.4. Combinación del algoritmo de recocido simulado con validación cruzada

Este algoritmo está concebido para encontrar los hiperparámetros óptimos o cercanos aplicando el metaheurístico de búsqueda de recocido simulado y empleando una validación cruzada como sistema de evaluación de la calidad del modelo producido con los mencionados hiperparámetros. La validación cruzada de k iteraciones (denominada en inglés *k-folds cross validation*) tiene como objeto garantizar que el modelo obtenido tenga una adecuada capacidad de generalización, entendiéndose como tal la habilidad del modelo para estimar correctamente el parámetro que se pretende aproximar cuando se emplean datos nuevos y diferentes de los empleados en la fase de entrenamiento del modelo. La validación cruzada de los modelos permite evitar en gran medida el sobreajuste que se produce cuando el modelo se entrena con una única tanda de datos.

La validación cruzada permite obtener modelos con capacidad de generalización que son independientes en gran medida de los datos empleados para su entrenamiento. Mediante esta técnica se fija el porcentaje de datos que se emplearán para el entrenamiento del modelo y los datos que servirán para la validación del mismo. En el caso de esta tesis se fijó el porcentaje de datos para entrenamiento en el 70% y el 30% para validación. Estos porcentajes se emplean habitualmente en el desarrollo de modelos de regresión o clasificación basados en sistemas de aprendizaje automático. Fijados los porcentajes, se procede a muestrear aleatoriamente k juegos de datos, cada uno de los cuales contiene todos los datos disponibles, el 70% en el grupo de entrenamiento y el 30% en el grupo de validación. Al objeto de evitar el sesgo en los datos, al final de este proceso ha de garantizarse que cada dato (el par constituido por la variable que se estima y las variables independientes) pertenece a uno y solamente a uno de los grupos de validación.

La incorporación de la validación cruzada al algoritmo de optimización de recocido simulado se produce en la forma de evaluar el error. Si en el caso precedente, la desviación del modelo era el error cuadrático medio obtenido con un solo modelo, con el enfoque arriba descrito, el error del modelo se computa como la media de los errores cuadráticos medios obtenidos cuando el modelo se entrena y se valida con cada uno de los k juegos de datos en que se ha dividido la población de muestra. De esta argumentación puede colegirse que la aplicación de la validación cruzada posiblemente produce modelos menos exactos en la estimación del valor de la variable independiente, pero más generalistas que si se construyesen con un solo juego de datos.

La figura 4.2 describe de forma gráfica el algoritmo combinado para la selección de los parámetros de los modelos de regresión que se emplearon en los estudios con *C. sorokiniana*. En todos los procesos de optimización de modelos de regresión se emplearon 10 iteraciones para la validación cruzada ($k = 10$). El algoritmo descrito fue programado en Pyhon 2.7.

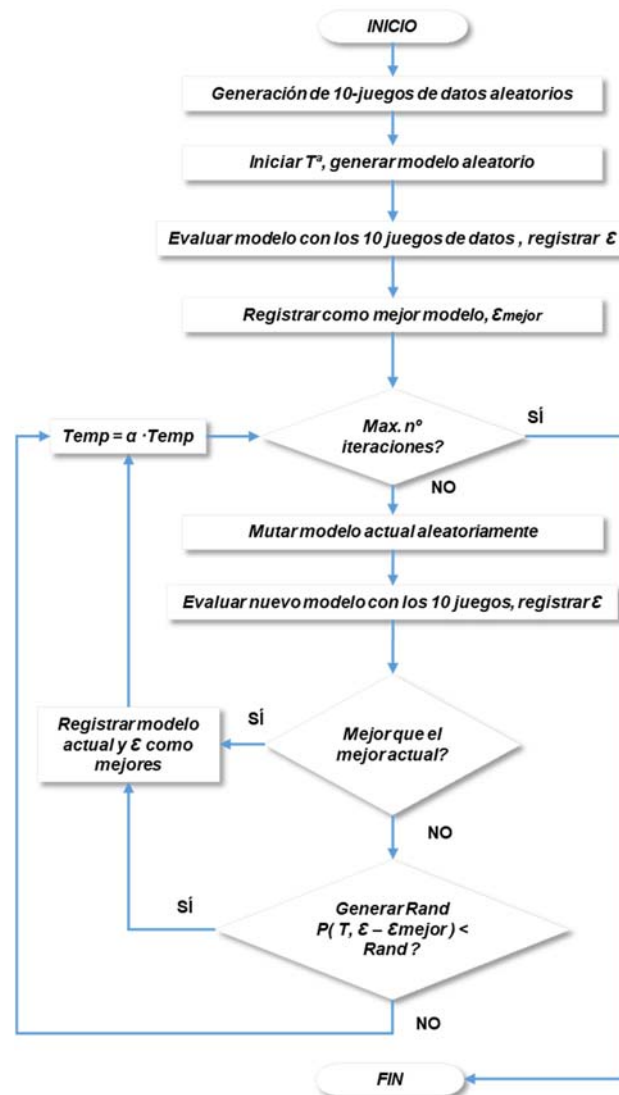


Figura 4.2. Algoritmo de optimización de la arquitectura de modelos de aprendizaje automático mediante la combinación de un algoritmo de optimización por recocido simulado y una validación cruzada

4.3. GENERACIÓN DE AGREGADOS FRACTALES VIRTUALES

Los agregados virtuales se generaron a través de una serie de *scripts* programados en Python 2.7 según un modelo de crecimiento agregado-agregado (*cluster-cluster aggregation* o CCA en inglés). Para el proceso de generación se diseñó e implementó en un algoritmo que combinaba otros dos algoritmos, uno anidado en el otro. El algoritmo anidado tuvo como objetivo lograr la combinación exitosa de pares de agregados, mientras que el algoritmo externo estaba destinado a *suministrar* los agregados que habrían de combinarse para llevar a cabo el proceso de crecimiento por CCA de forma iterativa y controlada hasta alcanzar el tamaño preestablecido para cada flóculo virtual. A continuación se describen los algoritmos mencionados.

4.3.1. Algoritmo para la combinación de pares de agregados

En un agregado de partículas primarias que tenga naturaleza fractal, el número de partículas primarias y el tamaño del flóculo se relacionan a través de la siguiente ley de potencia.

$$N = k_f \left(\frac{R_g}{a} \right)^{D_f} \quad (4.3)$$

En la ecuación 4.3, N es el número de partículas primarias, R_g es el diámetro de giro del agregado, a es el radio de las partículas primarias, D_f es la dimensión fractal y k_f el prefactor fractal. Al objeto de que la naturaleza fractal del agregado final a obtener sea igual a la consignada al inicio del proceso de generación, la ley de potencia ha de cumplirse en todas las etapas del proceso de crecimiento del flóculo (Skorupski et al., 2014).

Se seleccionó un modelo de agregación agregado- agregado como esquema de referencia para simular el crecimiento de los flóculos virtuales de microalgas. A pesar de ser un método mucho más complejo que el esquema de agregación partícula-agregado, el método seleccionado tiene una mejor capacidad para producir agregados similares a los observados experimentalmente, al contrario que los basados en agregaciones partícula-agregado que presentan simetrías y direcciones preferenciales de agregación.

El modelo de agregación construido en el curso del presente trabajo fue de tipo estocástico y se programó en Python 2.7. El modelo es capaz de garantizar que la ley de potencia, y con ella la naturaleza fractal del agregado, se verifica aproximadamente en todo el proceso de crecimiento del agregado virtual. El cumplimiento de este requisito geométrico se consiguió aplicando la ecuación 4.4 propuesta por Filippov et al. (2000), que relaciona las distancias de los centros de masas de dos agregados que coalescen con sus respectivas propiedades geométricas entendidas en términos de su dimensión fractal y radio de giro.

$$|\vec{\Gamma}|^2 = \frac{a^2(N_1+N_2)}{(N_1N_2)} \left(\frac{N_1+N_2}{k_f} \right)^{2/D_f} - \frac{N_1+N_2}{N_2} R_{g1} - \frac{N_1+N_2}{N_1} R_{g2} \quad (4.4)$$

Donde, Γ es la distancia entre los centros de masas de los agregados, a es el radio de las partículas primarias, N_1 y N_2 representan el número de partículas primarias de los agregados que se combinan, k_f es el prefactor fractal, D_f la dimensión fractal y R_{g1} , R_{g2} los respectivos radios de giro de los agregados que coalescen.

El radio de giro viene dado por la siguiente expresión.

$$R_g^2 = \frac{1}{N} \sum_{i=1}^N (r_i - r_0)^2 \quad (4.5)$$

Donde r_i representa el vector de posición de cada partícula primaria y r_0 la posición del centro de masas del agregado. La posición del centro de masas es la media de la posición de las partículas primarias, como se deduce de la siguiente expresión.

$$r_0 = \frac{1}{N} \sum_{i=1}^N r_i \quad (4.6)$$

El algoritmo de combinación de agregados tiene como objetivo final, conseguir la unión de los flóculos que coalescen cumpliendo dos premisas, siendo la primera que se cumpla la ecuación 4.4 y la segunda, que no haya partículas primarias que se solapen.

El algoritmo comienza con la verificación de que los agregados pueden efectivamente conectarse cumpliendo la ecuación 4.4. Esto se hace evaluando si la suma de las distancias de entre los centros de masas y las partículas más lejanas de cada agregado es mayor que L . Si esta condición se cumple, significa que si los agregados se sitúan a una distancia L y uno de ellos se rota, ambos agregados entrarán en contacto. De no cumplirse esta condición habrá que empezar el proceso empleando otros agregados iniciales.

Si los agregados son combinables, se produce la traslación de uno de ellos hasta que los centros de masas de los dos flóculos se encuentren a una distancia L . A continuación se elabora una lista para cada agregado en la que se incluyen las coordenadas de las células exteriores que son susceptibles de entrar en contacto con las del otro agregado, al objeto de reducir el esfuerzo computacional posterior.

Se seleccionan a continuación una partícula al azar de cada lista que serán las partículas a través de las cuales se producirá la unión de los dos agregados. En este punto han de realizarse dos rotaciones. La primera se hace tomando como eje de giro el vector \vec{r} , y tiene como objetivo situar la partícula seleccionada del segundo agregado, sobre el plano que forman los centros de masas de los dos flóculos con la partícula seleccionada del primer agregado, P_1 . A continuación se rota el segundo agregado para que las partículas seleccionadas al azar entren en contacto.

Una vez acabado el proceso de agregación tentativo, se emplearán ambas listas para verificar que no hay solapamiento entre partículas sin tener que verificar todas las células de los agregados, con el consiguiente ahorro en el tiempo de computación.

4.3.2. Algoritmo de crecimiento de agregados

El algoritmo descrito en la sección anterior se anida en el algoritmo de crecimiento de agregados. Este procedimiento permite la construcción de un agregado virtual consignando una D_f y un diámetro de Feret máximo. Dada la naturaleza estocástica del proceso de crecimiento y su carácter discreto, el DFM del agregado obtenido no es exactamente igual al valor consignado, si bien presenta valores muy cercanos dentro de una tolerancia fijada. El algoritmo de crecimiento está representado gráficamente en la figura 4.3 y, como puede observarse, tiene un carácter iterativo.

El proceso de crecimiento comienza con la generación de dos agregados, cada uno de los cuales compuesto por dos partículas primarias. Considerando que el proceso de crecimiento está en una fase intermedia de iteración, podemos colegir que el algoritmo trata de unir dos agregados genéricos (C1 y C2) y finaliza el proceso si el flóculo resultante tiene un DFM dentro del intervalo prescrito. Si el tamaño es mayor que el DFM de entrada más la tolerancia, el proceso se reinicia.

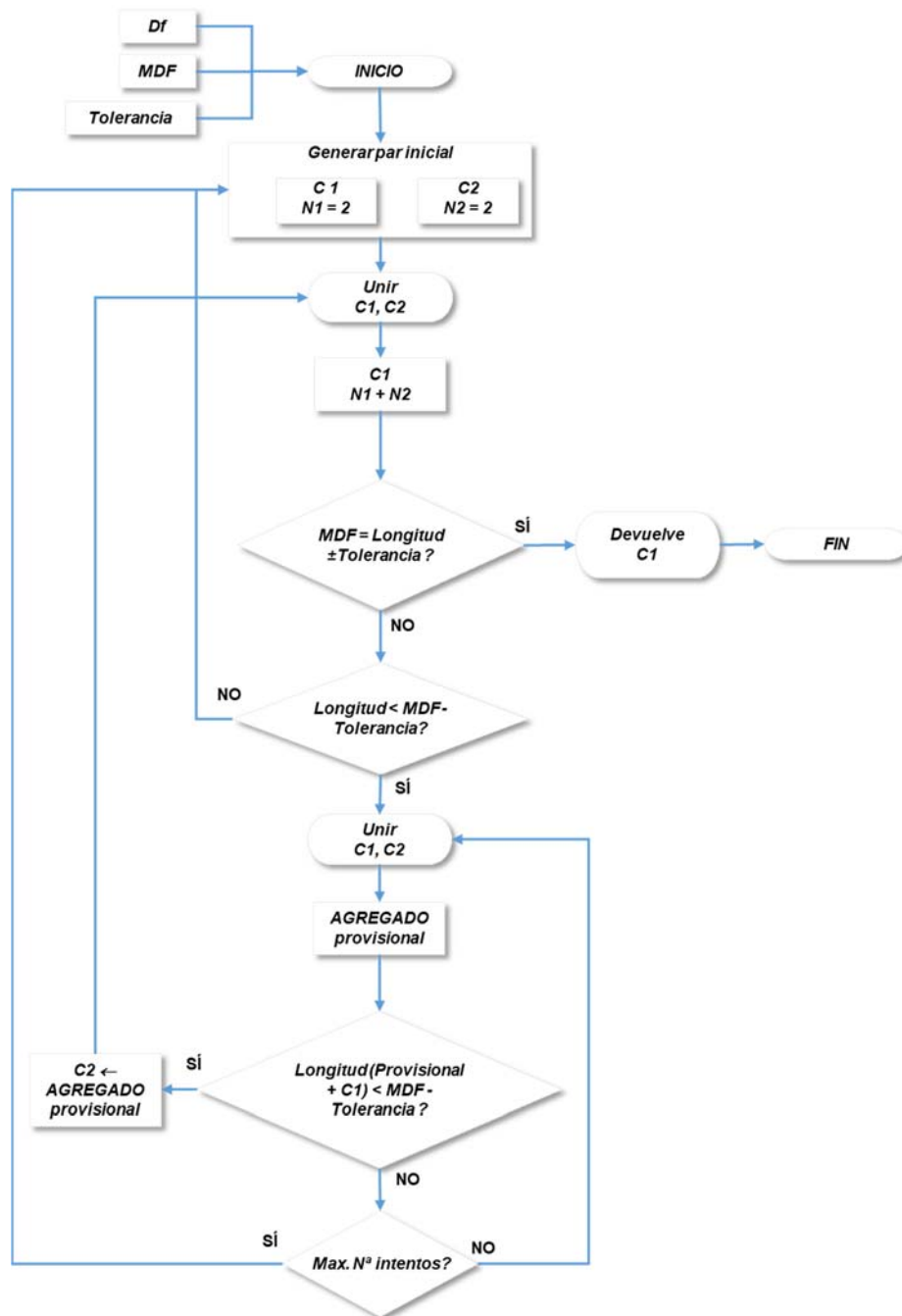


Figura 4.3. Algoritmo de crecimiento de agregados virtuales

Si el tamaño está por debajo de ese valor, el algoritmo intenta realizar una fusión posterior con uno de los clústeres genéricos anteriores. Si el tamaño del clúster recién formado (CPROV) es aceptable, se usa para comenzar un nuevo paso de fusión. Si el tamaño del clúster recién formado no es aceptable, el proceso de fusión anterior se lleva a cabo de nuevo hasta que el tamaño sea aceptable o se alcance el número máximo de intentos permitidos. Esta última etapa iterativa sirve para ahorrar tiempo de computación, ya que el sistema trata repetidamente de llevar a cabo la unión de dos agregados cuyo tamaño combinado tiene un alto potencial de ser aceptable, evitando así la necesidad de reiniciar el proceso de crecimiento.

4.4. ADQUISICIÓN DE DATOS DE LOS AGREGADOS FRACTALES VIRTUALES

4.4.1. Algoritmo de sedimentación de los agregados virtuales

Una de las tareas que se llevaron a cabo en este trabajo de investigación consistió en la comparación geométrica de las imágenes de flóculos de microalgas obtenidas por medio de microscopía láser confocal con las proyecciones de los agregados virtuales. Dado que los flóculos reales sedimentan en el fondo del portamuestras correspondiente, las imágenes obtenidas con el microscopio no serán del todo aleatoria, sino que corresponderán a posiciones estables de los flóculos, implicando éstas, en general, que el flóculo ofrece la mayor área de proyección al plano horizontal.

El desarrollo de un algoritmo de sedimentación virtual previo a la obtención de imágenes de los flóculos virtuales tuvo como objetivo precisamente simular el proceso de sedimentación a que se ven sometidos los agregados reales en la toma de imágenes microscópicas. Dado que es muy costoso computacionalmente encontrar el ángulo de proyección que se asocia a la máxima superficie, se implementó un algoritmo con el objetivo de realizar una estimación de la misma mediante una serie de rotaciones. La primera rotación conlleva asentar el agregado sobre el plano horizontal girándolo el ángulo correspondiente al comprendido entre dicho plano y el vector que une el centro de masas con su partícula primaria más lejana. Realizado este giro, se procede a practicar giros del flóculo sobre el eje formado entre la partícula más lejana, ya sobre el plano, y el centro de masas a razón de 1° cada vez hasta completar 360°, evaluándose el área en píxeles del flóculo proyectado sobre el plano horizontal y seleccionándose el ángulo que ofrece mayor área, para obtener la proyección del flóculo considerado. En cada caso, la proyección en el plano conllevó la generación de un mapa de bits provisional a través de OpenCV, una librería de visión artificial. En las matrices de proyección, cada elemento o píxel podía tomar dos valores, uno correspondiente a la proyección del agregado (valor 0) y otro correspondiente al fondo (valor 1). Dado que la escala de proyección se mantuvo constante, el número total de píxeles con valor 0 en cada imagen es una medida del área del agregado proyectado.

4.4.2. Estimación de la dimensión fractal en 2D de las proyecciones de agregados virtuales

Una vez estimada la posición del flóculo que resulta en una mayor área proyectada, el agregado virtual se proyectó efectivamente sobre el eje horizontal en forma de mapa de bits empleándose la mencionada librería. Las proyecciones generadas se almacenaron en archivos de imagen con formato PNG (*Portable Network Graphics*) para su posterior análisis.

La estimación de la dimensión fractal de las imágenes proyectadas se realizó por el método de conteo de celdas (*box counting*). Este método se basa en la relación invariante entre la dimensión y la escala en un objeto fractal (Li et al., 2006), que se puede expresar a través de la siguiente expresión:

$$N(\varepsilon) \propto \left(\frac{1}{\varepsilon}\right)^{D_f}, \quad (4.7)$$

Donde N es el número de unidades medidas del objeto fractal, ε es la escala a la que se efectúa la medida, y d es la dimensión de conteo de medidas. La dimensión fractal se calcularía mediante la siguiente ecuación:

$$D_f = \lim_{\varepsilon \rightarrow 0} \frac{\log N(\varepsilon)}{\log \frac{1}{\varepsilon}} \quad (4.8)$$

La anterior expresión sería válida para objetos fractales teóricos. Para fractales generados por ordenador y para objetos fractales naturales, existen limitaciones en el valor que puede tomar el factor de escala. En el caso que nos concierne, la menor escala que se puede manejar en el análisis de imagen es un píxel.

La estimación de la dimensión fractal proyectada por el método de conteo de celdas consiste en dividir la imagen del objeto en una retícula uniforme de celdas de longitud de arista ε y contar el número de celdas, N , ocupadas por el objeto, que se corresponderían con las unidades medidas de la ecuación anterior. Este conteo se repite sucesivamente reduciendo la longitud de la arista de las celdas de la retícula. La dimensión fractal, se obtendría como la pendiente de la siguiente recta:

$$\log N(\varepsilon) = k + D_f \log \frac{1}{\varepsilon} \quad (4.9)$$

El método de conteo de celdas se aplicó mediante la distribución FIJI del paquete de procesamiento de imágenes digitales ImageJ, a través del *plugin* BoneJ (Doubé et al., 2010).

4.4.3. Generación de espectros de tamaños de cuerda de los agregados virtuales

Al objeto de tener una población de muestra suficiente para el entrenamiento del algoritmo de estimación de la dimensión fractal a partir de datos de distribución de tamaños de suspensiones reales, se procedió a obtener espectros de tamaño de agregados virtuales. Las distribuciones de tamaño se generaron a partir de los flóculos virtuales. Cada flóculo virtual se sometió a 100 rotaciones aleatorias definidas a través de los correspondientes ángulos de giro aleatorios y vectores de rotación con origen en el centro de masas. Tras cada rotación, el agregado fue proyectado sobre el plano horizontal mediante la creación de un mapa de bits rectangular de dimensión un 20% superior a la máxima altura y anchura de la proyección de dicho flóculo. En la matriz de bits, se hicieron corresponder al fondo los píxeles con valor 0 y a los correspondientes al agregado se les asignó el valor 1.

Al objeto de simular el paso del haz de láser empleado en la adquisición de datos del tamaño de cuerda, sobre las matrices de proyección mencionadas se realizaron medidas verticales del número de píxeles contiguos con valor 1, que equivaldrían a la intersección del haz de láser con la superficie de las partículas reales. El número de píxeles contiguos con valor 1 se asoció pues a una longitud de cuerda que se almacenó en una lista. Estas longitudes de cuerda en píxeles se convirtieron en longitud a razón de 40 píxeles / 1,5 μm .

Las posiciones del haz virtual sobre la proyección del agregado se establecieron de manera aleatoria y separadas 40 píxeles entre sí. Con la idea de ahorrar esfuerzo computacional, solo se escanearon las áreas de la matriz de bits en las que se encontraba la proyección del flóculo

virtual en cada caso. Para simplificar el proceso, no se consideraron los posibles efectos que la curvatura de los agregados reales tendría sobre la medida de los tamaños.

Según lo descrito, cada rotación aleatoria daría lugar pues a una lista de tamaños de cuerda en micras. Para reproducir el formato de la distribución de tamaños suministrado por el software de la sonda láser empleada en la investigación, los tamaños de cuerda de las listas se ordenaron en 200 intervalos de tamaño organizados de manera logarítmica entre 1,021 μm y 4000 μm añadiendo el número de cuerdas de cada tamaño obtenidas al intervalo correspondiente.

A través de pruebas de rotación iniciales, se observó que a partir de 100 rotaciones aleatorias, las distribuciones de tamaño obtenidas con el programa no presentaban cambios, con lo que se decidió aplicar en cada caso este número de rotaciones aleatorias y no otro superior con el propósito de reducir el tiempo de computación.

Cada distribución de tamaños obtenida se sometió a un suavizado de señal mediante la aplicación de un filtro de mediana con una amplitud de 17 intervalos.

4.5. MÉTODOS PARA LA ESTIMACIÓN DE LA DIMENSIÓN FRACTAL MEDIA DE SUSPENSIONES REALES DE FLÓCULOS DE MICROALGAS

Las herramientas de modelado y algoritmos descritos en la sección anterior se emplearon como herramientas para el desarrollo de dos métodos de estimación de la dimensión fractal media de flóculos de biomasa microalgas en suspensión que se describen a continuación.

4.5.1. Método para la estimación de la dimensión fractal media de agregados de microalgas a partir del procesamiento de imágenes de microscopía láser confocal y modelado por ordenador

Este método, desarrollado en la presente tesis doctoral, se basa en la comparación de la geometría media de la proyección de los agregados reales con la de los flóculos generados por ordenador. En la figura 4.4 se muestra el esquema conceptual del proceso de estimación de la dimensión fractal media de los flóculos.

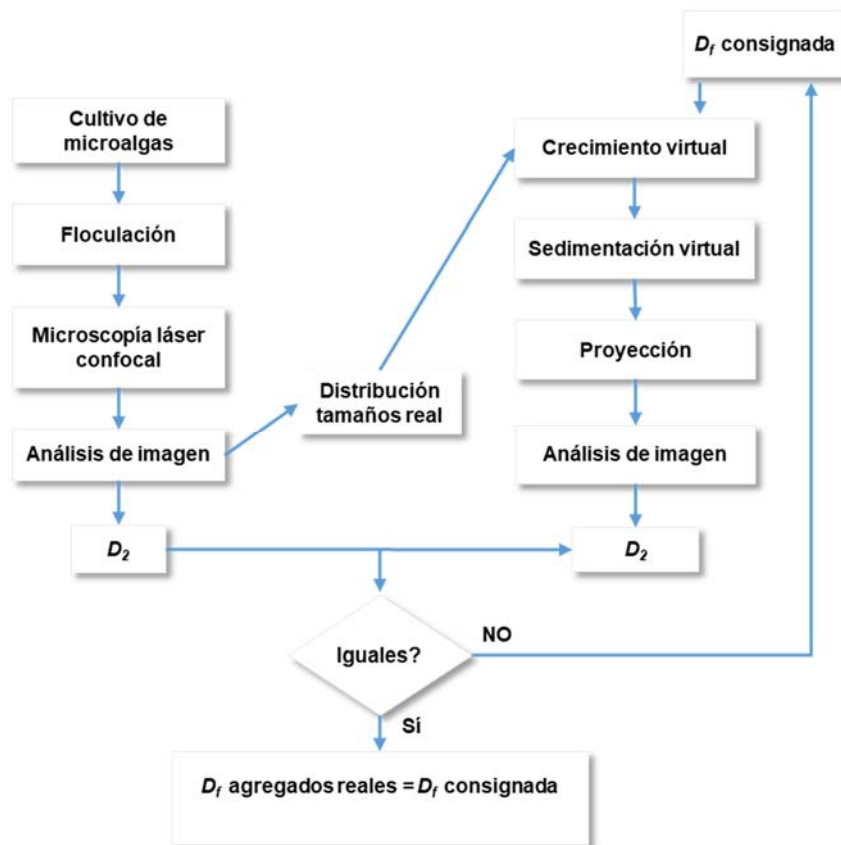


Figura 4.4. Esquema conceptual de la estimación de la D_f media de suspensiones de agregados por microscopía y modelado por ordenador.

Como se indicaba previamente, no existe una correspondencia unívoca entre la dimensión fractal de la proyección de un objeto fractal y su dimensión fractal en el espacio tridimensional. Esto implica que la dimensión fractal proyectada de un objeto no puede emplearse directamente en la estimación de D_f puesto que tanto las características de tal proyección como las naturaleza del objeto fractal (presencia de zonas solapadas, opacidad, etc.) van a afectar a la relación entre estos parámetros de forma particular. El enfoque propuesto en la metodología que se recoge en el diagrama anterior persigue precisamente reproducir, en la medida de lo posible, las condiciones particulares que presenta la relación entre D_2 y D_f en el caso de los flóculos de algas.

El proceso puede concebirse como dos subprocesos en paralelo, uno experimental y otro computacional.

El subproceso experimental consistió en inducir la floculación de las microalgas con quitosano en tres condiciones de agitación diferentes, estabilizando los flóculos formados con un dispersante y en la toma de muestras de los mismos para llevar a cabo adquisiciones de imágenes por medio de microscopía láser confocal.

Las imágenes obtenidas se procesaron por medio de la distribución FIJI del paquete de procesamiento de imágenes científicas ImageJ 1.151h. El procesado constó de las siguientes etapas:

- Proyección en el plano Z de las pilas de imágenes de cada muestra obtenidas en microscopía.

- Conversión de la imagen resultante en una de color de 8 bits.
- Aplicación de una modulación del contraste manual.
- Aplicación de un filtro de umbral automático.
- Conversión de la imagen resultante a modo binario.

A la imagen binaria obtenida se aplicó un script para seleccionar 120 flóculos de manera automática, separarlos en archivos individuales y almacenarlos en el directorio correspondiente a sus condiciones de floculación.

El diámetro Feret máximo se midió de manera automática con el software mencionado y el análisis de D_2 se llevó a cabo aplicando la herramienta “*Fractal Dimension*” del plugin BoneJ a la batería de archivos correspondiente a cada condición experimental. Esta herramienta de análisis se empleó utilizando 18 tamaños de caja diferentes de 200 píxeles a 6, con un cociente de reducción de 1.2.

Se estableció la D_2 media de la suspensión de agregados de microalgas como la media de las D_2 de los flóculos considerados. La D_2 estimada con este procedimiento se emplearía posteriormente para comparar con la D_2 estimada de los flóculos virtuales.

El subproceso computacional (parte derecha del diagrama en la figura 4.4) comienza con la generación de poblaciones de agregados virtuales concebidas para imitar las características de los agregados reales de microalgas obtenidos en los ensayos de floculación. De acuerdo al algoritmo de crecimiento de agregados presentado en secciones previas, los dos parámetros de entrada a fijar en el proceso son la dimensión fractal, D_f y el tamaño, este último en la forma del máximo diámetro de Feret.

Los diámetros de Feret medidos en cada condición de floculación se emplearon como tamaños de consigna para la generación de poblaciones de agregados virtuales constituidas por 120 agregados cada una. Se generaron 9 agregados para cada tamaño cada uno con una D_f de consigna entre 1,2 y 2,1 a razón de variaciones de 0,1. A cada flóculo virtual se le aplicó el algoritmo de sedimentación virtual y se obtuvo una proyección del mismo que se analizó de la misma forma que en el caso de las imágenes de flóculos reales, con la salvedad de que las proyecciones de flóculos virtuales se generaron ya binarizadas.

Asumiendo que los flóculos virtuales eran representativos de los reales, se infirió la dimensión fractal de los flóculos reales como la de consigna que permitía generar flóculos virtuales con una dimensión fractal proyectada similar a la medida en flóculos reales.

4.5.2. Metodología para el desarrollo del sistema de estimación de la dimensión fractal media de los agregados a través de un modelo de regresión por aprendizaje automático.

En la figura 4.5 se esquematiza el proceso de desarrollo del sistema de estimación de la dimensión fractal por medio de un modelo de regresión de aprendizaje automático. El enfoque planteado obedece a la necesidad de conseguir una población de muestras de variable predictora y variable objetivo lo suficientemente grande como para permitir el entrenamiento de un modelo de aprendizaje automático, que en última instancia, permita la estimación de

la dimensión fractal media de las suspensiones de flóculos reales mediante un modelo validado.

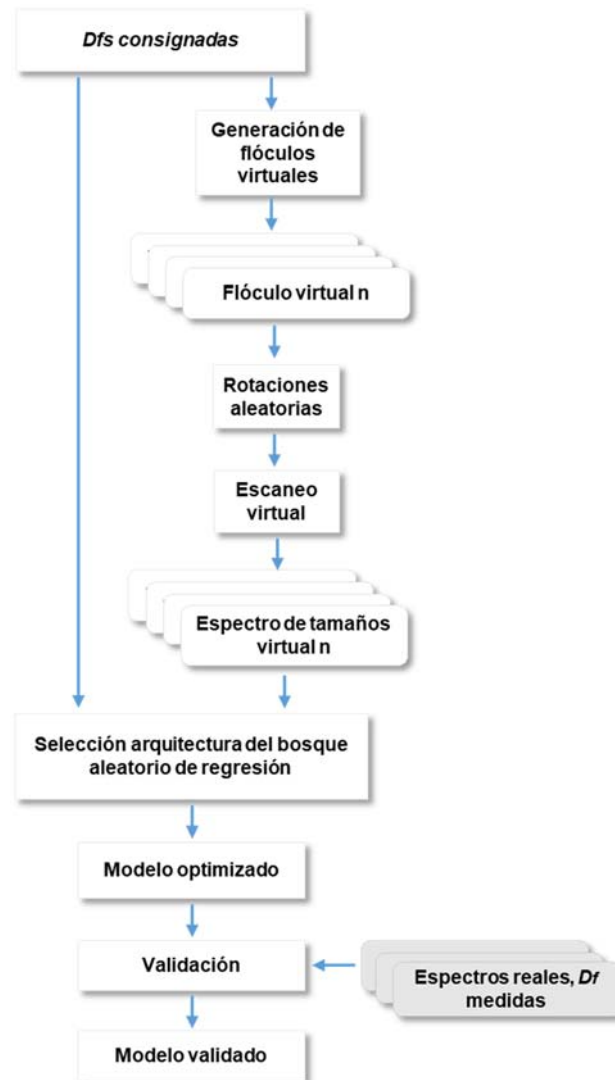


Figura 4.5. Procedimiento seguido para el desarrollo de un modelo de estimación de la D_f media de suspensiones de flóculos con distribuciones de tamaño.

El proceso comienza fijando una dimensión fractal y generando una serie de flóculos virtuales mediante los algoritmos de crecimiento anteriormente descritos. Cada agregado virtual se somete a rotaciones aleatorias de cada una de las cuales se obtiene una proyección en dos dimensiones que es objeto de escaneo virtual. Este último proceso permite obtener de cada proyección una distribución de tamaños. La combinación de las distribuciones de tamaño de las rotaciones aleatorias da como resultado la distribución de tamaños asociada al flóculo virtual considerado. Los pares constituidos por las distribuciones de tamaños de cuerdas y las respectivas dimensiones fractales de los agregados que generan tales distribuciones se emplearon como población de entrenamiento y selección de los hiperparámetros de un modelo de bosque aleatorio. Una vez se seleccionan los hiperparámetros óptimos del modelo de regresión, se procedió a la validación con datos reales de distribución de tamaños de cuerda de suspensiones de microalgas previamente floculadas de cuyas D_f habían sido estimadas a través del método descrito en la sección 3.5.1.

5

RESULTADOS

Los resultados obtenidos en la investigación desarrollada y su discusión se recogen en cinco publicaciones. A continuación se citan los títulos de las mismas y se resumen brevemente los resultados obtenidos en cada una de ellas.

- I. **Estimation of *Chlamydomonas reinhardtii* biomass concentration from chord length distribution data.** Lopez-Exposito, P., Suarez, A. B., & Negro, C. *Journal of applied phycology*, 28(4) (2016): 2315-2322.

En esta publicación se describen los resultados obtenidos en el desarrollo y aplicación de un sistema para estimar la concentración de biomasa en suspensiones de *C. reinhardtii* con distintos grados de agregación a través de la interpretación de datos de distribución de tamaños de cuerda con redes neuronales.

- II. **Laser reflectance measurement for the online monitoring of *Chlorella sorokiniana* biomass concentration.** Expósito, Patricio López, Angeles Blanco Suárez, and Carlos Negro Álvarez. *Journal of Biotechnology*, 243 (2017): 10-15.

En este trabajo se exponen los resultados obtenidos en el desarrollo de métodos de estimación de la concentración de suspensiones de *C. sorokiniana* con diferentes estados de agregación mediante la interpretación de datos de distribución de tamaños de cuerda por medio de dos sistemas de aprendizaje automático diferentes, los modelos de regresión por bosques aleatorios y las máquinas de vectores de soporte, ambos optimizados con una validación cruzada.

- III. **Microalgae harvesting by the novel flocculant hairy cationic nanocrystalline cellulose.** Patricio Lopez-Exposito, Cristina Campano, Theo G.M. van de Ven, Carlos Negro, Angeles Blanco. *Colloids and Surfaces B: Biointerfaces*, 178 (2019): 329-336.

Esta publicación describe el empleo de un nuevo tipo de floculante, la celulosa nanocristalina *hairy* cationizada, para la agregación de suspensiones de *Chlorella sorokiniana*. En el artículo se investiga el efecto de la dosis sobre el proceso de floculación y de las condiciones en que esta se realiza sobre la geometría y filtrabilidad de los agregados producidos.

- IV. **Estimating fractal dimension of microalgal flocs through confocal laser scanning microscopy and computer modelling.** Patricio Lopez Exposito, Angeles Blanco, and Carlos Negro. *Algal Research* 28 (2017): 74-79.

En esta publicación se describe el desarrollo y aplicación de un nuevo método de estimación de la dimensión fractal media de flóculos de microalgas. Este método se basa en comparar las dimensiones fractales en dos dimensiones de un conjunto de imágenes de microscopía láser confocal de flóculos de microalgas, con la dimensión fractal en dos dimensiones de conjuntos de flóculos generados por ordenador con tamaño y dimensiones fractales conocidas.

- V. **Direct estimation of microalgal flocs fractal dimension through laser reflectance and machine learning.** Lopez-Exposito, Patricio, Carlos Negro, and Angeles Blanco. *Algal Research*, 37 (2019): 240-247.

En este artículo se describe el desarrollo y validación de un nuevo método de estimación directa de la dimensión fractal media de una suspensión de flóculos de microalgas a través de la interpretación, con un modelo de regresión de bosques aleatorios, de los datos de tamaños de cuerda obtenidos de la suspensión por medio de una sonda por reflexión de haz enfocado. El conjunto de datos requerido para el entrenamiento y validación del modelo de bosques aleatorio se generó empleando flóculos virtuales generados por ordenador de dimensión fractal conocida, a cuyas proyecciones aleatorias se les aplicó un escaneo virtual simulando el comportamiento de la sonda FBRM. El modelo resultante del entrenamiento y validación con los datos virtuales se verificó con datos de tamaños de cuerda de suspensiones reales.

En la página siguiente se ofrece un resumen gráfico de las técnicas más relevantes empleadas en cada una de las publicaciones referidas más arriba. A continuación del mismo se ofrece un resumen que integra dichos resultados y su discusión.

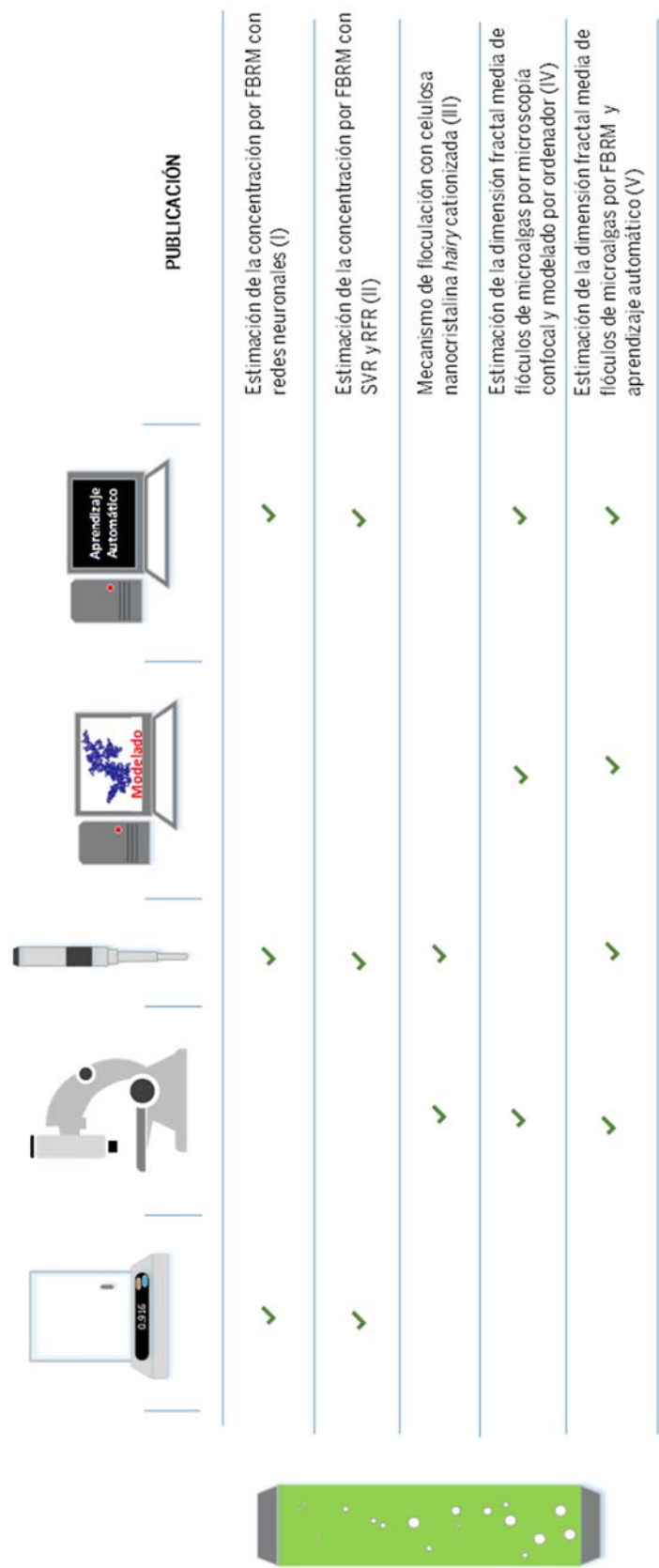


Figura 5.1. Resumen gráfico de las técnicas más relevantes empleadas en las publicaciones asociadas a la presente tesis.

5.1. MÉTODOS PARA LA ESTIMACIÓN DE LA CONCENTRACIÓN DE LA BIOMASA DE MICROALGAS

5.1.1. Estimación de la concentración de *Chlamydomonas reinhardtii* por medio de redes neuronales artificiales.

El desarrollo de modelos basados en redes neuronales tipo perceptrón requiere, como se ha mencionado en el apartado 4 de esta memoria, de una serie de datos de entrenamiento que permitan al modelo aprender la relación entre los datos de tamaños de cuerda de la suspensión y la concentración de biomasa. Para generar la población de datos necesaria para este propósito se emplearon 15 suspensiones de partida, con concentraciones de biomasa de *C. reinhardtii* en peso seco comprendidas entre 0,2 y 2,7 g·L⁻¹. Estas suspensiones se floccularon parcialmente con diferentes dosis de quitosano, como se indica en la sección de métodos, y se sometieron a un análisis con la sonda M500L FBRM durante 6 minutos tomando un dato de tamaños de cuerda cada 10 segundos. Del software del equipo se extrajeron 20 vectores de tamaños de cuerda consecutivos. Las parejas de datos integradas por la distribución de tamaños de cuerda (predictor) y concentración de biomasa (variable estimada) se emplearon en el entrenamiento y selección de la arquitectura óptima del modelo de perceptrón.

Con la arquitectura de perceptrón seleccionada a través de la optimización de los hiperparámetros del modelo, se consiguieron valores relativamente bajos en el error de predicción, alcanzándose un error cuadrático medio de 0,0309 (g·L⁻¹)² superadas las 18.000 iteraciones. Con estos datos, la correlación entre los datos reales medidos y los predichos por el modelo fue significativamente alta con un R² = 0,923 (figura 5.2).

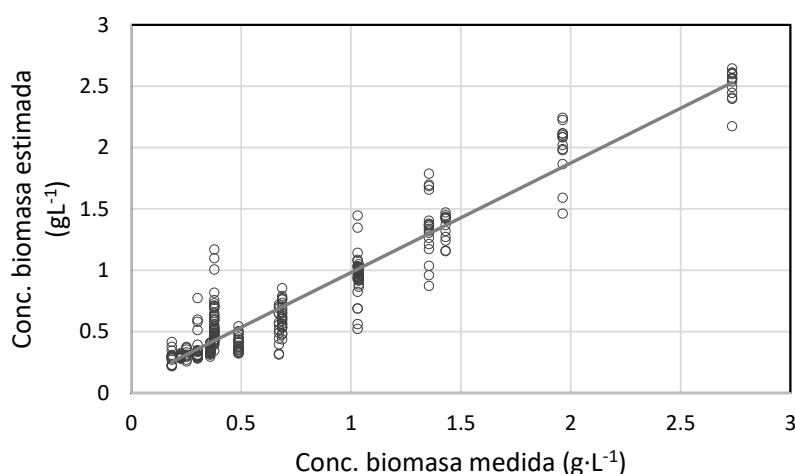


Figura 5.2. Concentración de biomasa estimada con el modelo de perceptrón frente a la medida en los datos de validación.

La tabla 5.1 muestra los valores seleccionados para los hiperparámetros considerados en la optimización.

Tabla 5.1.- Hiperparámetros del perceptrón multicapa seleccionado

Elemento	Valor
Número de capas ocultas	2
Número de neuronas en la 1ª capa oculta	18
Número de neuronas en la 2ª capa oculta	14
Sesgo en la primera capa oculta	No
Sesgo en la segunda capa oculta	Sí
Sesgo en la capa de salida	Sí
Valor inicial de los pesos que van desde la entrada hasta la primera capa oculta	0,1
Valor inicial de los sesgos en capas ocultas	1
Intervalo de valores iniciales de los pesos que van desde la primera capa oculta hasta la segunda capa oculta o salida	1
Función de transferencia de la primera capa oculta	Tangente hiperbólica
Función de transferencia de la segunda capa oculta	Logística
Función de transferencia de la capa de salida	Tangente hiperbólica
Relación de los errores cuadráticos medios y los pesos cuadrados medios (regularización)	0,01

5.1.2. Estimación de la concentración de la biomasa de *Chlorella sorokiniana* floculada por medio de bosques aleatorios y máquinas de vectores de soporte

En el desarrollo de modelos de estimación con redes neuronales, los modelos de regresión por bosques aleatorios y los basados en máquinas de vectores de soporte de la sección siguiente, también se requiere de una población de datos para llevar a cabo el entrenamiento y validación de los modelos. En estos dos casos, la adquisición de datos se llevó a cabo midiendo a través de la sonda G400 las distribuciones de tamaño de cuerda de 19 suspensiones de microalgas floculadas con distintas dosis de quitosano con concentraciones de biomasa seca en el intervalo de 0,06 a 2,2 gL⁻¹. En cada caso, la medida se llevó a cabo durante 7 minutos, adquiriendo un tamaño de cuerda de la suspensión cada 10 segundos. De los 42 vectores de tamaño adquiridos se descartaron el primero y el último y los restantes 39 se emplearon en el entrenamiento del modelo.

Como se comentaba en la sección de métodos, se empleó el algoritmo de recocido simulado en combinación con un esquema de validación cruzada con 10 subconjuntos al objeto de garantizar la capacidad de generalización de los correspondientes modelos.

Empleando un modelo de correlación por bosques aleatorios, la correlación obtenida entre las concentraciones reales y estimadas como promedio de los 10 subgrupos de validación fue de $R^2 = 0,81$ (figura 5.3) con un error cuadrático medio de 0,08 (gL⁻¹)², que se alcanzó en la iteración 5.525. En el modelo de bosques aleatorios seleccionado se observa una tendencia a subestimar los valores de concentración de biomasa seca para suspensiones de concentración por encima de 1,0 gL⁻¹.

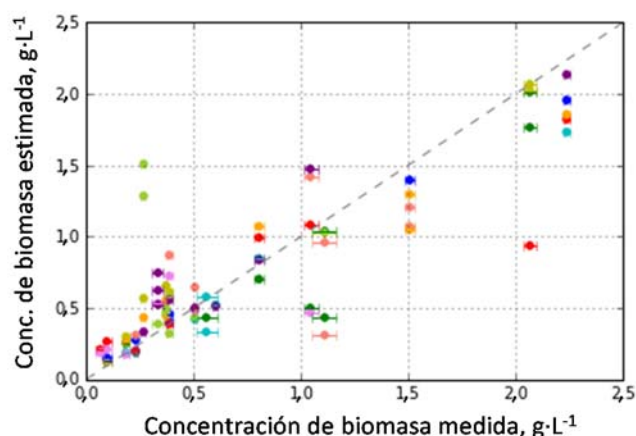


Figura 5.3. Concentración de biomasa estimada con el modelo de bosques aleatorios frente a la medida en los datos de validación.

Los valores de los hiperparámetros del modelo seleccionado se recogen en la tabla 5.2.

Tabla 5.2 - Hiperparámetros del bosque aleatorio de regresión seleccionado

Hiperparámetro	Valor
Número de árboles	525
Profundidad máxima	9
Mínimo número de muestras permitido	3
Máximo número de intervalos de tamaño a dividir	74

En el caso del modelo de predicción por máquinas de vectores de soporte, la correlación obtenida entre los datos de concentración medidos y las concentraciones estimadas fue algo mejor, $R^2=0,87$ (figura 5.4). El mínimo ECM ($0,087 \text{ (g L}^{-1})^2$) se alcanzó en el ciclo 4.631, si bien el proceso de selección requirió un mayor tiempo de ejecución del algoritmo de optimización que en el caso anterior. En este caso, la tendencia a subestimar la concentración a concentraciones superiores a $1,0 \text{ gL}^{-1}$ no fue tan pronunciada como en el caso anterior. Los valores de los hiper-parámetros del modelo seleccionado se recogen en la tabla 5.3.

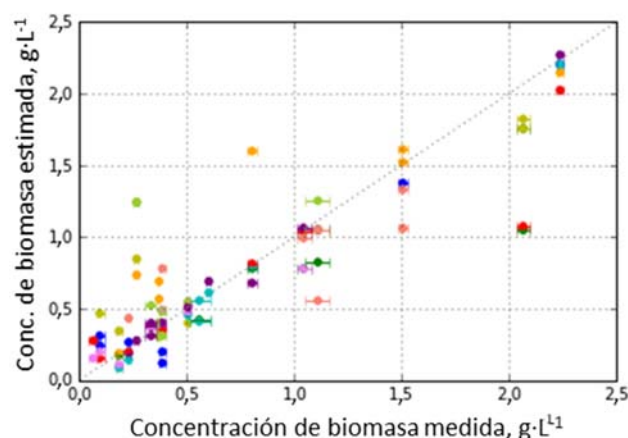


Figura 5.4. Concentración de biomasa estimada con el modelo de máquinas de soporte de vectores frente a la medida en los datos de validación.

Tabla 5.3 - Hiperparámetros del modelo de regresión con máquinas de vectores de soporte.

Hiperparámetro	Valor
Tipo de kernel	Función de base radial
C	60
ϵ	0,02
Gamma	0,05

5.2. INFLUENCIA DE GEOMETRÍA DE LOS AGREGADOS DE MICROALGAS EN LA FILTRABILIDAD DE LAS SUSPENSIONES DE BIOMASA FLOCULADA.

Los resultados que se resumen en esta sección se corresponden con los descritos en el artículo número III.

5.2.1. Floculación de suspensiones de *Chlorella sorokiniana* con CNC-HC

En los ensayos de floculación se demostró que el nuevo floculante basado en CNC-HC era efectivo en la floculación de biomasa de microalgas en un intervalo de dosis alrededor del punto isoeléctrico. Como cabría esperar, el tamaño medio de agregado obtenido fue tanto mayor cuanto más suaves eran las condiciones de agitación aplicadas, si bien, con la velocidad de agitación más baja, el tamaño medio de cuerda de los flóculos generados en dosis media (27,5 ppm) y alta (55 ppm) presentaba oscilaciones amplias, no observándose crecimiento de los flóculos a dosis baja (5 ppm). Las mencionadas oscilaciones posiblemente eran debidas a la inestabilidad de los flóculos formados y al paso por la ventana de la sonda de flóculos de tamaño grande de forma semiperiódica.

A velocidades de agitación superiores, 200 rpm y 300 rpm, se observó floculación con la dosis más baja y las dosis media y alta presentaron mayor estabilidad que en el caso anterior.

En la fase de refloculación, los tamaños medios de cuerda medidos en las suspensiones medias y altas presentaron valores muy similares a pesar de que los tamaños medios de los agregados fueran distintos en la fase de floculación. En vista de los valores de la eficacia de la floculación calculados, se concluyó que la floculación de biomasa de microalgas no ocurría por neutralización de cargas como mecanismo principal, sino a través de otro mecanismo controlado por la velocidad y tipo de adsorción de los nanocristales sobre la superficie de las microalgas.

5.2.2 Efecto de la dosis de CNC-HC sobre el potencial Z de la muestra

La eficacia de la floculación en términos de reducción de la absorbancia y la relación entre el potencial zeta y la dosis de CNC-HC aplicada se muestran en la figura 5.5. La eliminación de absorbancia máxima se observó con la dosis de CNC-HC de $16,5 \text{ mg}\cdot\text{L}^{-1}$, un 86,5% con potencial Z de $-6,31 \text{ mV}$. Este valor fue similar a los referidos por Gerde et al. (2014) en la floculación de *Scenedesmus spp.* empleando almidón cationizado con mayores dosis de floculante.

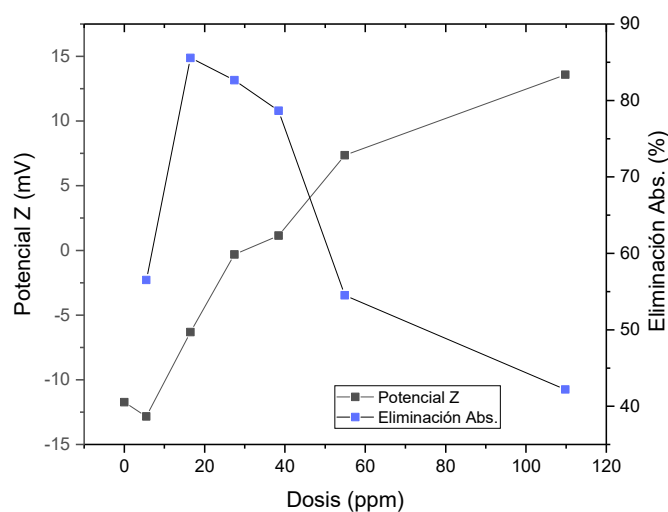


Figura 5.5. Potencial-Z y reducción de absorbancia en función de la dosis de CNCC

El punto isoeléctrico (PIE) se encontró entre las dosis de $27,5$ y $38 \text{ mg}\cdot\text{L}^{-1}$. En este intervalo, la eliminación de la absorbancia fue bastante similar, 79 y 82% respectivamente, lo que significa que la dosis óptima en términos de clarificación del agua se encuentra cerca del PIE. Es de destacar el hecho de que la amplia diferencia en la cantidad de floculante añadido en las dos dosis anteriormente referidas no se tradujo en un cambio equivalente del potencial-Z, lo que apunta a que la floculación no se produjo exclusivamente por neutralización de cargas. Para dosis muy por encima del PIE, los valores de eliminación de la absorbancia fueron significativamente más bajos, 54,5% para $55,5 \text{ mg}\cdot\text{L}^{-1}$ (potencial-Z $+7,36 \text{ mV}$) y 42,2% en el ensayo realizado con la dosis de $110 \text{ mg}\cdot\text{L}^{-1}$ (potencial Z $+13,59 \text{ mV}$).

5.2.3 Efecto de la dosis de CNC-HC sobre la morfología de los agregados

Con la estimación de la dimensión fractal proyectada media, se observó que, fijadas las condiciones de agitación, la morfología de los flóculos, y por tanto su mecanismo de formación, era dependiente de concentración de nanocrisales empleada como agente floculante. La figura 5.6 recoge varias imágenes microscópicas de muestras de flóculos correspondientes a la fase de floculación y a la de refloculación.

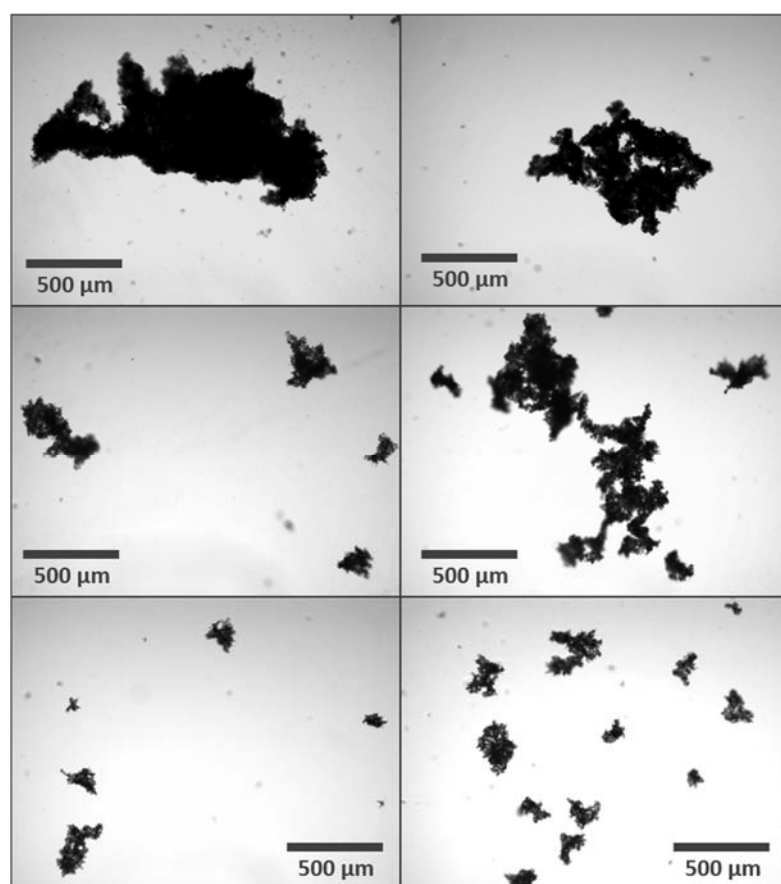


Figura 5.6. Imágenes microscópicas de suspensiones de *C. sorokiniana* floculadas con 27,5 ppm de CNC-HC aplicando velocidades de agitación de a) 100 rpm; b) 200 rpm; y c) 300 rpm. Las imágenes de la izquierda corresponden a la fase de floculación y las de la derecha a la de refloculación.

Como se observa en la figura 5.6, el flóculo inicial obtenido a 100 rpm exhibe una mayor opacidad que el de la fase de refloculación, que presenta huecos en su estructura. En el caso de 200 rpm, los agregados de la fase de floculación parecen de menor tamaño que los de refloculación, y estos últimos parecen estar formados por la unión de flóculos más pequeños. La observación del ensayo a 200 rpm es extensible a las imágenes de los ensayos de floculación inicial a 300 rpm.

La figura 5.7 recoge los valores de D_2 medios estimados para las suspensiones obtenidas por floculación aplicando las tres intensidades de agitación consideradas. La principal conclusión que se puede extraer de los datos representados es que para una misma dosis de CNC-HC,

una mayor intensidad de agitación se traduce en la producción de flóculos de dimensión fractal menor. En la literatura se ha descrito tanto este fenómeno (Moruzzi et al., 2017; Wyatt et al., 2013), como el opuesto (Chakraborti et al., 2003; Li et al., 2007b), es decir, la obtención de flóculos más compactos a mayores velocidades de agitación. En el caso que nos concierne, es posible que la mayor apertura de los agregados producidos a mayor agitación se deba a la naturaleza semirrígida de las células de microalgas que por un lado limita la capacidad de reconfiguración de los flóculos, y por otro, hace que la difusión de las células hacia el interior del flóculo sea más lenta que la adhesión de células sobre la superficie exterior del mismo.

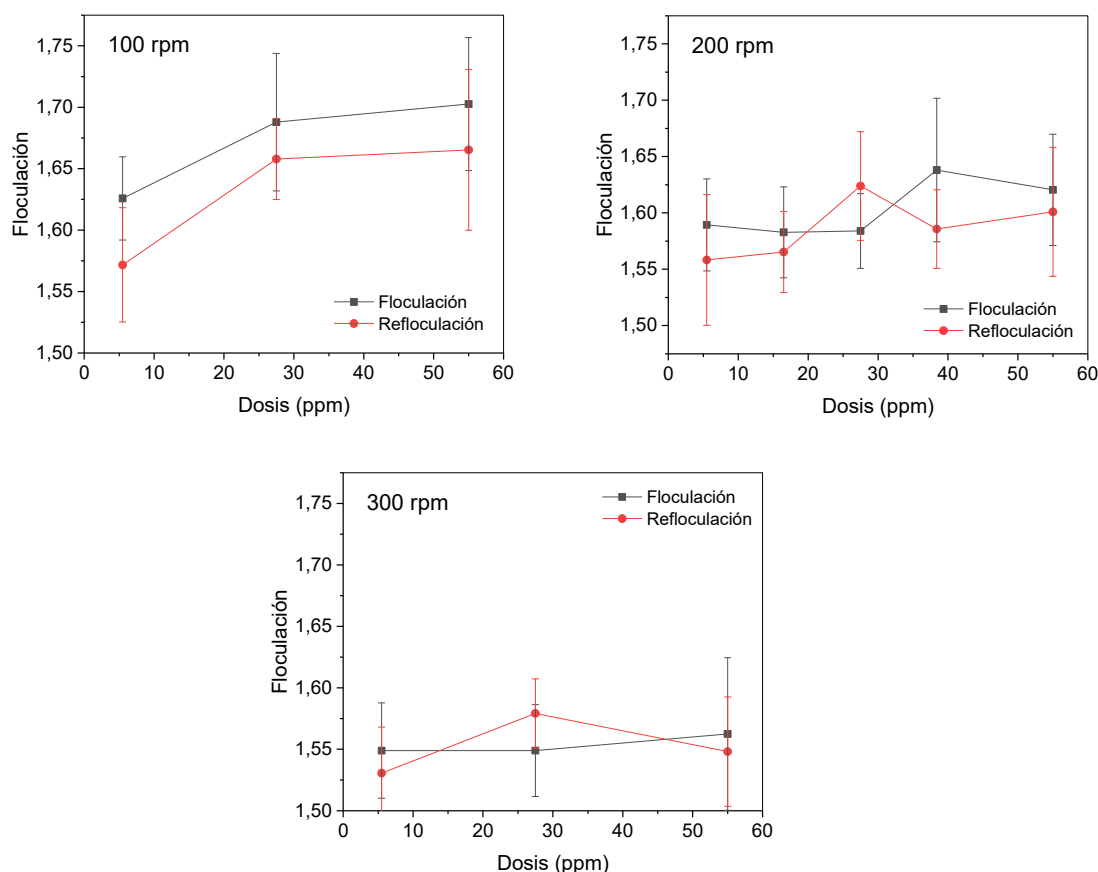


Figura 5.7. D_2 en función de la dosis para las velocidades de agitación consideradas.

Con estos datos, y en virtud de la geometría y distribución de carga de los nanocrisales de CNC-HC, se propusieron dos mecanismos para explicar la formación de flóculos de microalgas con este floculante en función de la dosis empleada.

Mecanismos de unión intercelular mediados por nanopartículas de CNC-HC

Dada la morfología particular de la CNC-HC y su distribución de carga, son posibles tres configuraciones principales de unión entre microalgas dependiendo de la concentración de nanopartículas, tal y como se muestra en la Figura 5.8 a, b y c. Teniendo en cuenta la concentración de CNC-HC en el PIE, 27,5 mg L⁻¹ de CNC-HC está presente por cada 0,4 g L⁻¹ de microalgas, lo que corresponde a 0,12 g de biomasa seca por litro, suponiendo un contenido de humedad del 70% en las células. Asumiendo que las células de microalgas son

esferas de 3 μm de diámetro, el área específica de la suspensión sería de $0,8 \text{ m}^2 \text{ L}^{-1}$. Asumiendo que los nanocristales de CNC-HC son cilindros de diámetro 5 nm y longitud de 380 nm, el área proyectada específica de la dosis indicada sería de unos $24 \text{ m}^2 \text{ L}^{-1}$. Estas estimaciones indican que en la suspensión que se flocula existe una cantidad de CNC-HC mucho mayor de la necesaria para formar una monocapa de nanocristales depositados a lo largo de su eje neutro (tangencialmente) sobre las algas. Si las partículas de CNC-HC se depositaran erguidas (ortogonalmente) sobre la superficie de las células, un área de $0,8 \text{ m}^2$ podría albergar 471 mg de CNC-HC, es decir, aproximadamente 17 veces la dosis de CNC-HC en el PIE. Dado que es poco probable que todas las partículas de CNC-HC se depositen ortogonalmente, podemos concluir que en el PIE, hay una combinación de partículas de CNC-HC tumbadas y erguidas sobre las células de microalga, como se muestra en la figura 5.8, en la que la escala del nanocrystal ha sido exagerada para facilitar la percepción del esquema.

Para concentraciones bajas de CNC-HC o en la etapa inicial de floculación, la unión de CNC-HC en células de microalgas comenzará con la adsorción de nanopartículas tangencialmente en la superficie de una microalga. En esta configuración, parte de la carga positiva de los extremos amorfos del nanocrystal se neutralizará por la carga negativa de la superficie celular. Esas áreas en la superficie celular libres de nanocristales adsorbidos se cubrirán rápidamente con más partículas de CNC-HC absorbidas en una configuración plana, con solo una pequeña porción de nanocristales adsorbidos verticalmente a través de uno de sus extremos positivos (figura 5.8a). La unión entre las células en este escenario probablemente será mediada por una combinación de nanocristales adsorbidos de manera plana (ya que la repulsión electrostática entre las regiones cristalinas de las partículas de CNC-HC adsorbidas de manera plana es despreciable, debido a que el espesor de la doble capa ($\sim 1 \text{ nm}$) es menor que el diámetro de varillas) y las depositadas ortogonalmente. Cabría esperar que las uniones intercelulares en este caso sean relativamente débiles debido a la escasa presencia de nanopartículas ortogonales y la resistencia relativamente baja de las uniones planas, dado que la carga disponible en los extremos de los nanocristales está parcialmente neutralizada. La carga positiva de nanocristales adsorbidos tangencialmente constituye un obstáculo para la adsorción de nuevas nanopartículas en la superficie celular debido a las repulsiones electrostáticas. Además, dado que el eje central de nanocristales es neutro, solo será posible una cobertura parcial de la superficie celular a través de la deposición tangencial aleatoria. Como resultado, una célula cubierta hasta el grado máximo posible con partículas planas de CNC-HC aún retendrá parte de su carga negativa.

En concentraciones cercanas al PIE, un aumento de la cobertura de nanopartículas planas en la superficie de las células implicará impedimentos estéricos y electrostáticos que favorecerán gradualmente la adsorción ortogonal de nanocristales en puntos libres, como se muestra en la figura 5.8b. Las células se unirán entre sí principalmente mediante nanocristales adsorbidos ortogonalmente, como se muestra en las figuras 5.8b y c. Dado que la adsorción ortogonal en la superficie de las células tiene lugar a un ritmo menor que la adsorción plana inicial, los flóculos formados en estas condiciones también retendrán cierta carga negativa en su interior.

La estructura del agregado resultante del crecimiento a través de este mecanismo se esquematiza en la Figura 5.8d. La parte externa del agregado en contacto con el medio está altamente cubierta con CNC-HC, tanto en configuraciones tangenciales como ortogonales, pero la unión entre las células en esta región se debe principalmente a estas últimas. Dentro de los flóculos, las células probablemente estarán menos cubiertas por los nanocristales, debido a la pequeña probabilidad de que las partículas individuales penetren en los flóculos puesto que la difusión es lenta en comparación con el movimiento inducido por el flujo en el exterior. Esto implica que la región interna de los flóculos presente una carga negativa neta y, probablemente, una estructura menos compacta que las regiones externas.

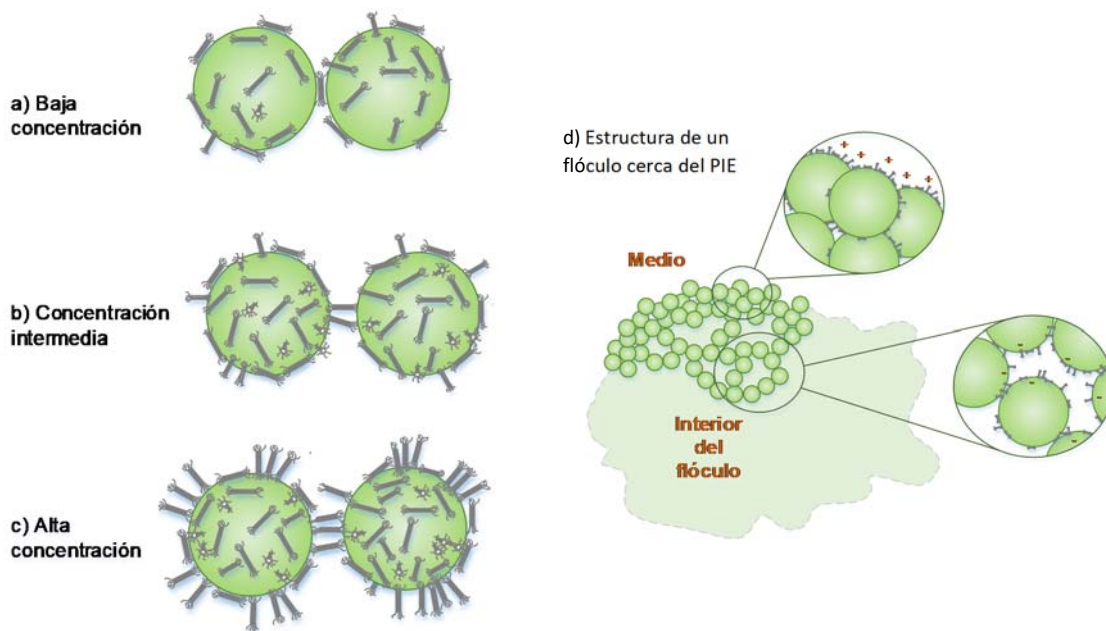


Figura 5.8. Posible unión mediada por CNC-HC intercelular en función de la concentración de nanopartículas: a) baja, b) media, c) alta concentración y d) estructura de un flóculo de microalga inducida con una dosis de CNC-HC cerca del PIE.

En virtud a la proporción entre las áreas específicas de las células y los CNC-HC, es de esperar que en todas las concentraciones de CNC-HC que se consideran en los ensayos la cobertura de las células con nanocristales adsorbidos tangencialmente sea máxima. Por otro lado, para altas concentraciones de CNC-HC, la superficie de las células estará probablemente poblada con gran cantidad de nanopartículas adsorbidas ortogonalmente. Los flóculos formados en estas condiciones serán más rígidos que en los casos anteriores y presentarán una menor carga neta negativa en su interior.

El mecanismo propuesto parece concordar con los resultados experimentales obtenidos. Las dos configuraciones de adsorción propuestas, siendo la ortogonal la etapa limitante, explicarían el retraso en la floculación observada a bajas revoluciones si se supone que la floculación comienza cuando la adsorción vertical de los nanocristales se torna relevante. Además, la formación de flóculos que tienen una carga negativa neta interna podría explicar las pequeñas diferencias en el potencial zeta observadas cerca del PIE cuando las dosis de 27,5 y 38,5 ppm.

5.2.4 Efecto de la morfología sobre la filtrabilidad de las suspensiones de microalgas floculadas

Los ensayos de filtración se llevaron a cabo con la dosis de $27,5 \text{ mg L}^{-1}$, considerada óptima, aplicando las tres condiciones de agitación aplicadas previamente. Se consideraron tres presiones de vacío diferentes, 0,5, 0,65 y 0,83 bar. La figura 5.9 muestra el volumen de cultivo filtrado con el tiempo obtenido con la aplicación de las tres presiones de vacío. La pendiente de la curva en cada gráfica representa la velocidad de filtración alcanzada con las condiciones experimentales correspondientes.

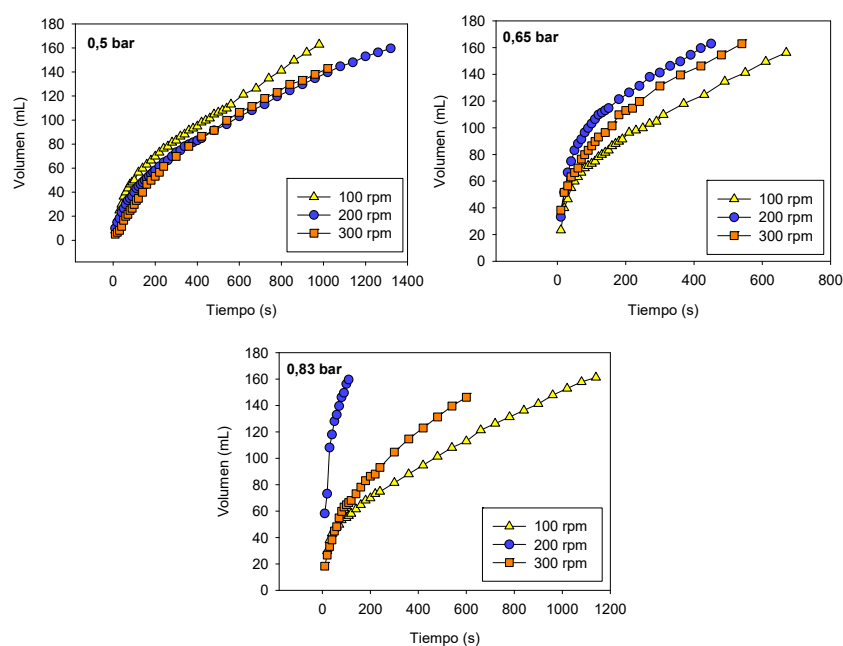


Figure 5.9. Volumen de filtrado frente al tiempo para las tres presiones de vacío consideradas: a) 0.5 bar, b) 0.65 bar y c) 0.8 bar; y d) reducción de absorbancia en función de la presión de vacío.

En la prueba realizada a -0.5 bar, la velocidad de filtración inicial más alta fue la del cultivo floculado a 100 rpm (unos $0,58 \text{ mL} \cdot \text{min}^{-1}$), siendo la filtración más lenta la del cultivo floculado a 300 rpm ($0,26 \text{ mL} \cdot \text{min}^{-1}$). Después de la etapa de filtración inicial, las tres velocidades de filtración alcanzaron un comportamiento relativamente constante con un valor similar, siendo la muestra de 100 rpm la más rápida en alcanzar una velocidad de filtración constante. A esta baja presión de vacío, la velocidad de filtración parece verse favorecida por una mayor compacidad de los flóculos, ya que la filtración fue más rápida con la muestra de 100 rpm ($D_2 = 1.68$, LMC promediado = $147 \mu\text{m}$ en la meseta de la fase de floculación). Aunque este hecho parece estar en conflicto con la idea de que los flóculos de dimensiones fractales menores contribuyen a la permeabilidad, dado que el vacío aplicado es de baja intensidad es probable que la torta de filtración no esté comprimida, en cuyo caso, la porosidad entre los flóculos se ve favorecida por flóculos más grandes como observaron Lee et al. (2003). El patrón similar de velocidad de filtración observado en las muestras de 200 y 300 rpm indica, para esta presión de vacío, que D_2 tiene una mayor influencia en la

filtrabilidad que el tamaño del flóculo, dado que ambas muestras presentaron valores de D_2 muy cercanos (1.58 y 1.54 respectivamente) y LMC muy distintos, un 37% más grande en el caso de la muestra de 200 rpm (LMC promedio = 135 μm) que el medido a 300 rpm (LMC promedio = 85 μm). En todos los casos, la eliminación de la turbidez del filtrado en comparación con el ensayo realizado al 0% de CNC-HC fue superior al 98% (figura 5.9), lo que implica que la floculación seguida de la filtración fue efectiva para recoger la biomasa de microalgas a la presión de vacío aplicada.

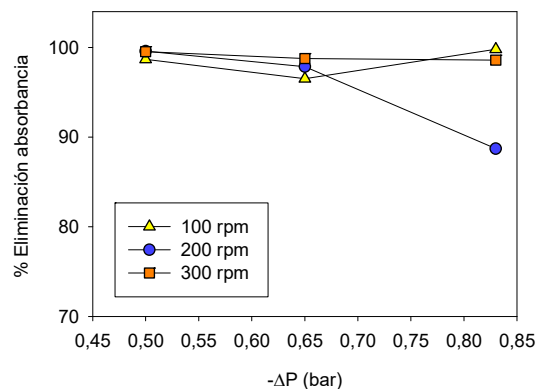


Figure 5.10. Eliminación de la absorbancia a distintas presiones de vacío en función de la velocidad de agitación en la floculación.

Cuando se empleó una presión de vacío de 0,65 bar, se observó que la filtración se veía favorecida por dimensiones fractales bajas, lo cual se deduce de la mayor velocidad de filtración de las muestras de 200 y 300 rpm respecto a la floculada a 100 rpm. En este caso, la mayor longitud promedio de flóculo de la suspensión floculada a 200 rpm permitió una filtración de la muestra ligeramente más rápida que la observada para la muestra agregada a 300 rpm. La obtención de mayor rendimiento de filtración mediante la combinación de tamaños de flóculo mayores y D_2 más bajas está de acuerdo con los resultados descritos en estudios similares (Amjad et al., 2015; Park et al., 2006). Esta situación corresponde a la formación de una torta con una porosidad intra-flocular muy alta que permite un flujo de agua relativamente fácil a través de la misma. A una presión de vacío de 0,65 bar, la eliminación de la turbidez del filtrado fue superior al 96%. La filtración más lenta observada en la muestra floculada a 100 rpm se puede atribuir a la compresión de la torta de filtración. Dado que los flóculos que integran la torta presentan una porosidad intra-flocular más baja como consecuencia de su mayor D_2 , la compresión de la torta reduciría la porosidad inter-flocular que resultaría en el detrimento de la porosidad de la torta. A la presión de vacío máxima ensayada (0,8 bar), el efecto del tamaño sobre la velocidad de filtración para las suspensiones de D_2 más bajo se hizo más evidente que en las pruebas de vacío intermedio. En este caso, la suspensión floculada a 200 rpm presentó una velocidad de filtración notablemente alta tanto en la fase inicial como en la de filtración constante. El efecto de que los tamaños más grandes favorezcan la porosidad de la torta a altas presiones de vacío en flóculos de baja dimensión fractal se observó también en Lee et al. (2003). La rápida filtración de la muestra de 200 rpm correspondió, sin embargo, a una eliminación de turbidez

del 88,7%, la más pobre de cuantas se observaron en los ensayos de filtración. La presencia de turbidez en el filtrado se puede explicar cómo la consecuencia de la acción de un flujo de líquido intenso producido por la alta presión de vacío sobre una torta de alta porosidad. Este flujo produciría una presión dinámica negativa capaz de desgarrar algunos agregados de la torta y drenarlos a través del medio de filtración. A pesar de que la D_2 medida en la suspensión de 300 rpm fue similar a la de la muestra de 200 rpm, no se observó que la biomasa pasara al filtrado, muy probablemente debido a su menor LMC, que condujo a la formación de una torta más compacta con una mayor resistencia al paso del agua.

Tanto en las muestras floculadas a 100 rpm, como a 300 rpm, el aumento de la presión de vacío conllevó peores velocidades de filtración en las fases iniciales y constantes que en los ensayos con un vacío más suave. Nuevamente a esta presión, la filtración de la suspensión floculada a 300 rpm fue más rápida que la inducida a 100 rpm. La eliminación de la absorbancia observada en los casos de 100 y 300 rpm fue de 99,8 y 98,6% respectivamente.

5.3. MÉTODO DE ESTIMACIÓN DE LA DIMENSIÓN FRACTAL EN 3D DE FLÓCULOS DE MICROALGAS PROYECCIÓN MEDIANTE MICROSCOPIA LASER CONFOCAL Y MODELADO POR ORDENADOR.

5.3.1. Adquisición de imágenes microscópicas de suspensiones de agregados de microalgas y su caracterización morfológica.

Las imágenes de microscopía láser confocal fueron obtenidas tras la floculación con quitosano y fijación de los flóculos con dispersante en tres condiciones de agitación. A modo de ejemplo, en la figura 5.11 se muestran tres imágenes obtenidas por microscopía láser confocal de suspensiones de *C. sorokiniana* floculadas a 200, 350 y 500 rpm respectivamente. Estas imágenes son las resultantes de proyectar todas las capas de imagen obtenidas en cada caso sobre el plano empleando el método de máxima intensidad.

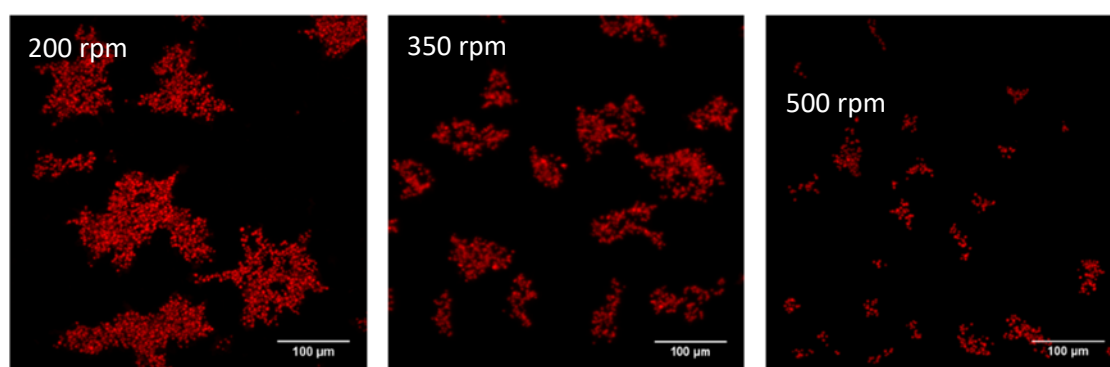


Figura 5.11. Imágenes de microalgas floculadas a distintas velocidades de agitación obtenidas por microscopía láser confocal

El análisis cualitativo de las imágenes de la Figura 5.11 permite reconocer que los agregados obtenidos a menor velocidad son los que presentan mayor tamaño y compacidad, siendo los producidos con la intensidad de agitación mayor, los que presentan menor tamaño y un carácter más abierto.

En la figura 5.12 se muestran varios ejemplos del resultado de procesar los agregados de las imágenes obtenidas en microscopía confocal.



Figura 5.12. Imágenes binarizadas de agregados de *C. sorokiniana* floculados a 200, 350 y 500 rpm.

En la figura se aprecia claramente la diferencia de tamaño de los flocúlos formados en función de las condiciones de agitación aplicadas en el momento de la floculación, siendo mayores en las condiciones de operación más suaves. Asimismo, se observa que las condiciones de agitación parecían influir en la estructura que presentan los agregados, siendo más compacta en las condiciones de agitación más suaves y más elongada cuando la floculación se llevaba a cabo con mayores intensidades de agitación, como se comprobaría cuantitativamente mediante la determinación de la dimensión fractal proyectada.

La dimensión fractal proyectada de los agregados analizados en función de su tamaño se representa en la figura 5.13. Las dimensiones fractales proyectadas medias se recogen en la tabla 5.4. Se observa que las dimensiones fractales proyectadas de las suspensiones floculadas a 200 y 350 rpm presentan valores similares, siendo menor la media de la suspensión floculada a 500 rpm. Los valores medidos son compatibles con los descritos en la literatura en ensayos similares (Wyatt et al., 2013; Yuheng et al., 2011).

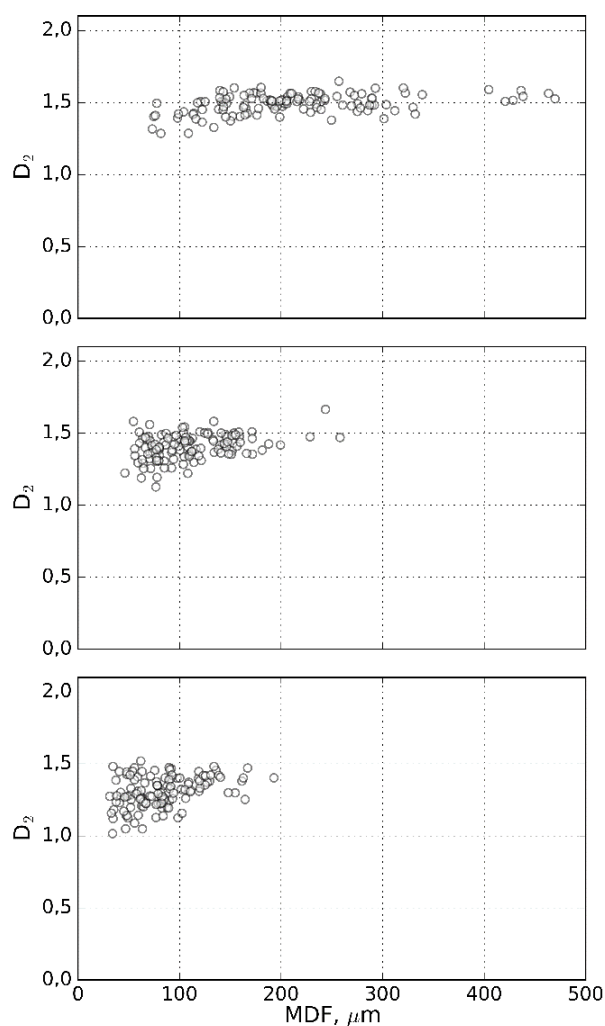


Figura 5.13. Dimensión fractal proyectada en función del tamaño de agregados obtenidos a a) 200, b) 350 y c) 500 rpm.

Cabe destacar en este punto, que las diferencias cuantitativamente pequeñas en los valores de la dimensión fractal proyectada se traducen en divergencias sustancialmente mayores en la dimensión fractal tridimensional. La menor compacidad de los flóculos de microalgas observada cuando la floculación se produce a mayores intensidades de agitación se ha observado previamente en la literatura (Dong et al., 2014; Wyatt et al., 2013), como se explica en la sección 5.2.2 de esta memoria.

Tabla 5.4.- Dimensiones fractales proyectadas promedio (D_2) obtenidas en las condiciones de agitación consideradas.

Velocidad de agitación (rpm)	D_2 media
200	1,49 ($\pm 0,07$)
350	1,40 ($\pm 0,09$)
500	1,31 ($\pm 0,11$)

Estas dimensiones fractales proyectadas medias se compararán posteriormente con las obtenidas de los flóculos virtuales para inferir la dimensión fractal en 3D de los flóculos reales.

5.3.2. Generación de flóculos virtuales con D_f y máximo diámetro de Feret consignados

La Figura 5.14 muestra una serie de agregados virtuales generados con MDF 307 μm y dimensiones fractales entre 1,2 a 2,1.

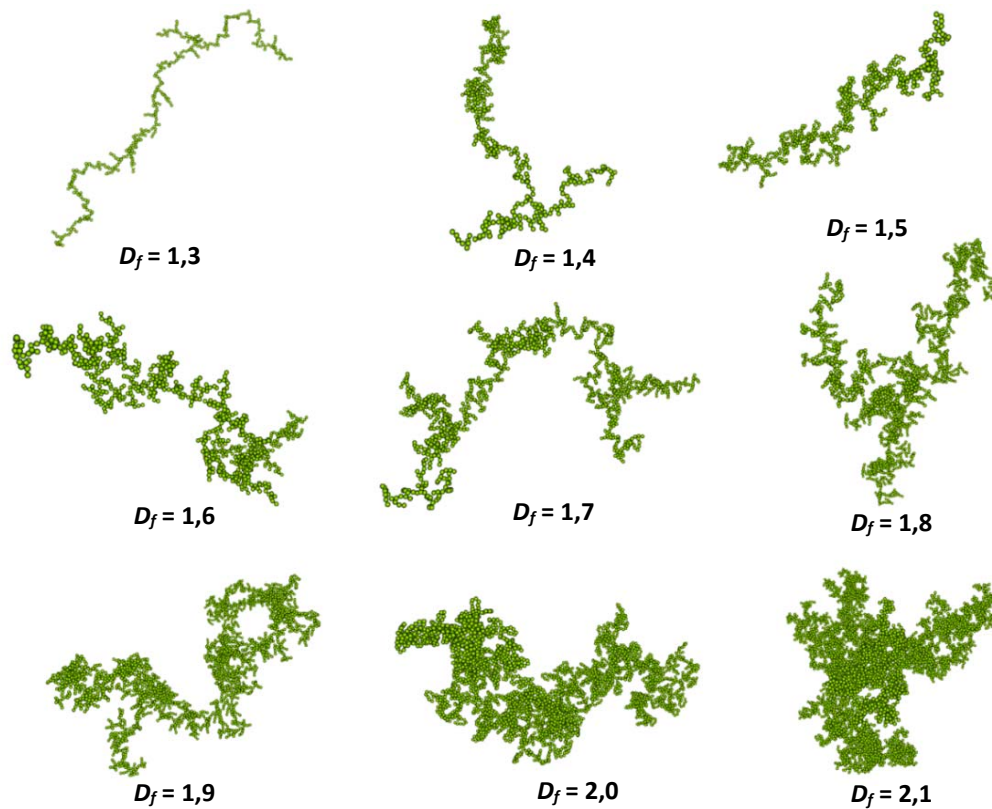


Figura 5.14. Agregados virtuales con MDF 307 μm y D_f de consigna entre 1,2 y 2,1.

La selección de dicho intervalo de dimensiones fractales se basó en la literatura relacionada con las investigaciones sobre la dimensión fractal tridimensional de los sistemas agregados formados por partículas primarias de tamaño constante y relativamente monodispersas que emplearon sistemas de floculación similares a los aplicados en el presente trabajo (Jiang and Logan, 1996; Logan and Kilps, 1995).

Los agregados virtuales se generaron en tres familias, cada una de las cuales correspondiente a una de las tres condiciones de agitación consideradas en esta sección de la investigación, es decir, a 200, 350 y 500 rpm. A cada condición de agitación correspondía una distribución de MDF, que se utilizaría como tamaño de consigna. Para cada familia de MDF se generaron nueve grupos de 120 flóculos (9 dimensiones fractales y 120 MDF).

La generación de flóculos virtuales con D_f y MDF aplicando el algoritmo de crecimiento descrito en la sección de métodos resultó efectiva, consiguiéndose agregados virtuales cuyo tamaño se aproximó significativamente al de consigna. La figura 5.15 muestra la relación entre todos los MDF reales (las tres familias) y el MDF medio resultante de promediar los tamaños de los flóculos virtuales obtenidos empleando el MDF real como tamaño de consigna para las nueve D_f de consigna consideradas ($R=0,99$, desviación media con respecto al MDF consignado = $\pm 2,5 \mu\text{m}$).

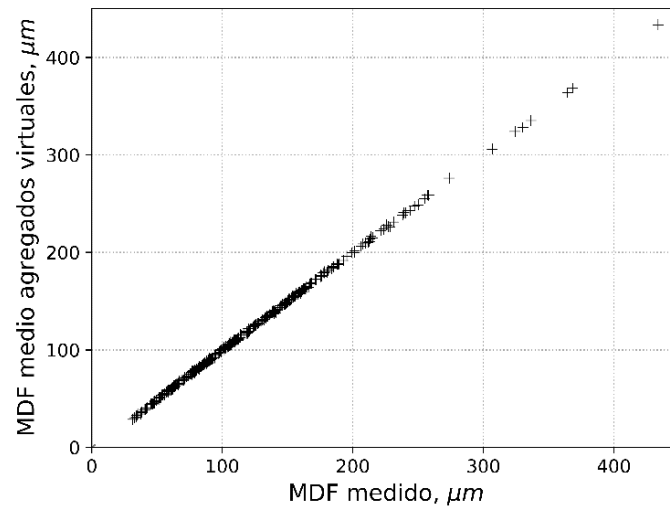


Figura 5.15. DFM de los agregados virtuales en función del DFM medidos

La figura 5.16 muestra la relación entre el número de partículas primarias en cada agregado virtual generado con su dimensión fractal de consigna y MDF. En la gráfica puede apreciarse, especialmente en las dimensiones fractales mayores, que el número de partículas primarias de los agregados presenta una relación potencial con el respecto a la longitud característica de los flóculos.

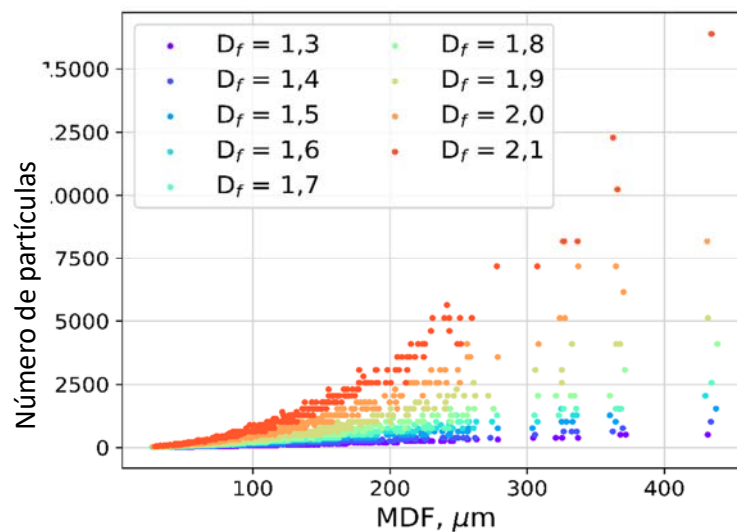


Figura 5.16. Número de partículas en el agregado en función de su MDF de consigna.

5.3.3. Proyección de flóculos virtuales y estimación del área proyectada máxima

Dado que los agregados virtuales se generaban de forma aleatoria y sin una orientación preferencial en el espacio, se hacía necesario “sedimentarlos virtualmente” para poder comparar sus proyecciones con las imágenes microscópicas de flóculos reales. La figura 5.17 muestra cuatro proyecciones aleatorias de un flóculo virtual de $D_f = 2,1$ y MDF 107 μm .

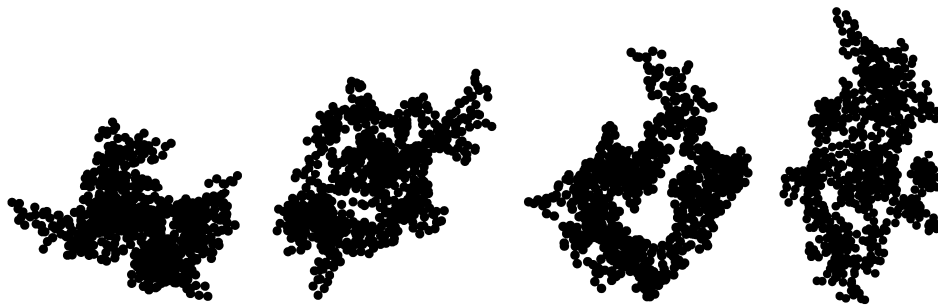


Figura 5.17. Cuatro proyecciones aleatorias de un agregado virtual con $D_f = 2,1$ y MDF 107

El algoritmo de sedimentación virtual de flóculos permitió obtener las proyecciones horizontales de los agregados virtuales con mayor posibilidad de ocurrencia si estos existieran en el dominio real a través de la estimación del área de apoyo máxima. La figura 5.18 muestra tres proyecciones del mismo agregado una vez se ha asentado su eje mayor sobre el plano Z. Las proyecciones corresponden a distintos ángulos de rotación sobre dicho eje. La primera proyección presenta la de mayor área de apoyo, a la segunda le corresponde un área intermedia y a la tercera, la menor área de entre los ángulos considerados.

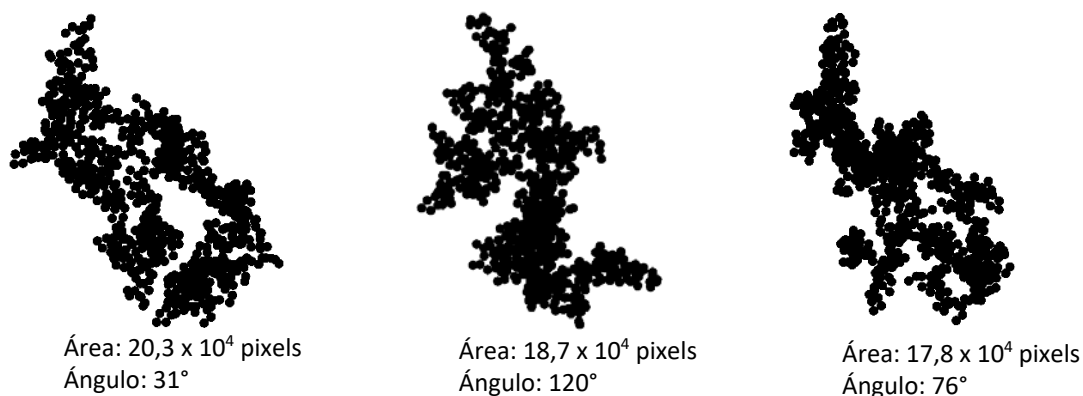


Figura 5.18. Proyecciones de un agregado virtual con $D_f = 2,1$ y MDF 107 tras la rotación alrededor de su eje máximo.

Al objeto de ofrecer un ejemplo de la variabilidad en la proyección de los flóculos, la figura 5.19, muestra la variación del área de proyección del flóculo de la figura anterior en función del ángulo de rotación una vez que el eje mayor de la partícula se ha rotado para que esté sobre el plano Z. A partir de la estimación del área máxima de proyección en cada flóculo se

seleccionó el ángulo de rotación correspondiente y la proyección asociada se empleó para la estimación de la D_2 .

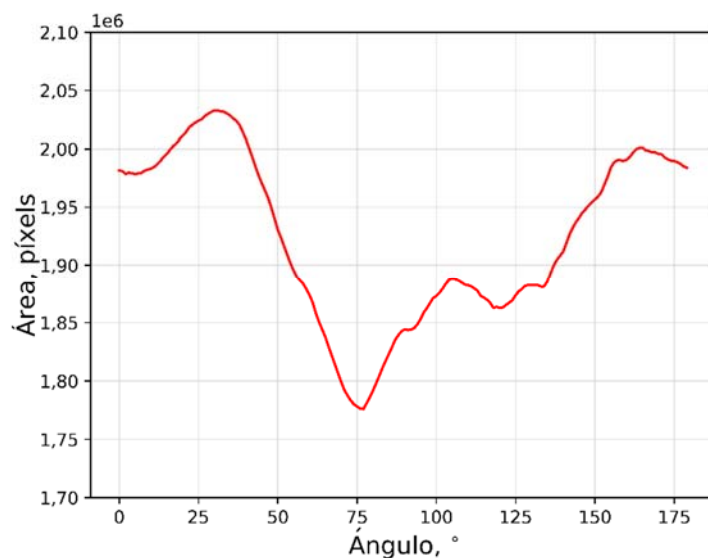


Figura 5.19. Área de la proyección del floculo con $D_f = 2,1$ y MDF $107 \mu\text{m}$ en función del ángulo de giro.

5.3.4. Inferencia de la D_f por comparación entre dimensiones fractales proyectadas.

Los resultados de la medición de la D_2 de los agregados se recogen en la figura 5.20 para cada uno de las familias de agregados.

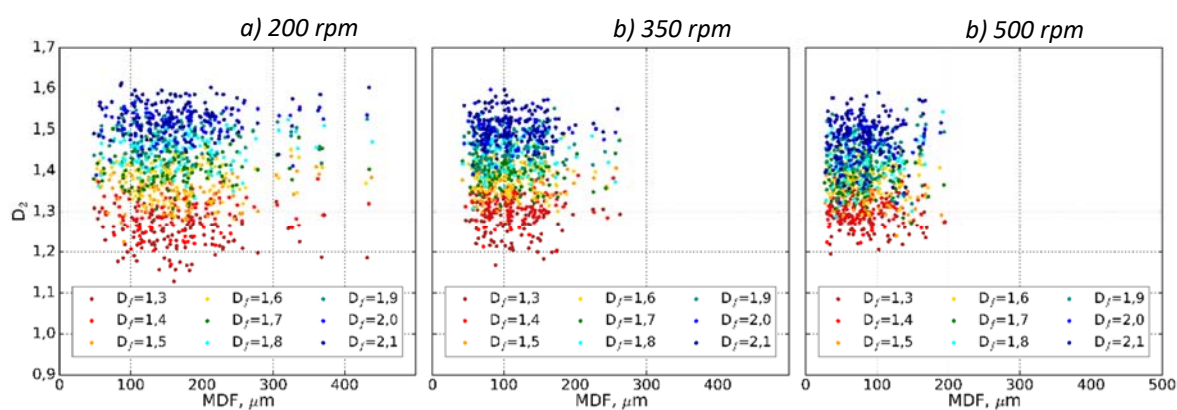


Figura 5.20. D_2 de cada agregado virtual para las familias a) 200, b) 350 y c) 500 rpm

La media de los valores obtenidos en cada familia y dimensión fractal de consigna se recoge en la tabla 5.5. Se observa que los valores de la dimensión fractal presenta una ligera dependencia con el tamaño medio de los floculos, apreciándose una diferencia más notable entre la D_2 media de los floculos de la familia de 500 rpm y las otras dos (200 y 350 rpm) que entre estas últimas. Este fenómeno fue observado anteriormente por Jullien et al. (1994) en la proyección de agregados fractales y puede deberse a que los agregados más pequeños están compuestos por un número relativamente bajo de partículas primarias cuya estructura estaría cercana al límite de lo que sería un objeto fractal.

Tabla 5.5. Dimensiones fractales proyectadas promedio (D_2) obtenidas en cada familia de agregados para las dimensiones fractales de consigna consideradas.

	Familia 200 rpm		Familia 350 rpm		Familia 500 rpm	
D_f consigna	D_2	Desv.	D_2	Desv.	D_2	Desv.
1,3	1,24	0,04	1,27	0,04	1,29	0,03
1,4	1,29	0,04	1,31	0,04	1,31	0,03
1,5	1,35	0,04	1,37	0,03	1,37	0,04
1,6	1,39	0,04	1,40	0,04	1,39	0,04
1,7	1,41	0,04	1,40	0,04	1,39	0,04
1,8	1,45	0,04	1,44	0,04	1,42	0,05
1,9	1,48	0,04	1,47	0,04	1,44	0,05
2,0	1,52	0,03	1,50	0,03	1,47	0,05
2,1	1,53	0,04	1,51	0,04	1,48	0,05

Por comparación con los valores recogidos en la tabla 5.4 de la sección 4.3.1, se obtuvo que a las suspensiones floculadas a 200, 350 and 500 rpm les correspondían dimensiones fractales medias de 1,9, 1,7 and 1,4 respectivamente.

5.4. DESARROLLO DE UN MODELO DE ESTIMACIÓN DE LA D_f A PARTIR DE DATOS DE TAMAÑO DE CUERDA DE SUSPENSIONES DE MICROALGAS.

5.4.1 Obtención de tamaños de cuerda de flóculos virtuales

La aplicación de rotaciones aleatorias a cada flóculo y el escaneo virtual al que se sometieron las correspondientes proyecciones permitió la producción de una población de distribuciones de tamaño de cuerda sin procesar. La figura 5.21 muestra una las distribuciones de tamaño de cuerda obtenidas a partir de cada una de las rotaciones aleatorias a las que este se ha sometido.

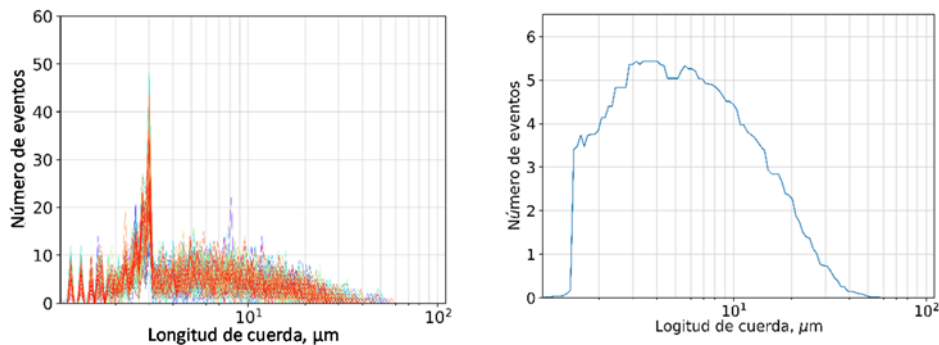


Figura 5.21. Longitudes de cuerda sin procesar de 100 rotaciones de un flóculo virtual y la misma tras someterla a filtrado y promediado.

El pico que se observa en la gráfica corresponde aproximadamente al tamaño de las partículas primarias. En las distribuciones de tamaño de agregados reales de microalgas obtenidas por medio de la sonda de reflexión por haz enfocado no se observan este tipo de picos. La discrepancia entre la simulación y el resultado de la sonda se debe probablemente a dos efectos. Por un lado, la simulación realiza una detección exacta de píxeles proyectados mientras que la sonda emplea la intensidad del láser reflejado para estimar los tamaños, con lo que posiblemente no detecta bien las discontinuidades si estas son del orden del límite de detección de la sonda (aproximadamente $1\ \mu\text{m}$).

Por otro lado, el software de adquisición de la sonda cuenta con un sistema de filtrado para corregir señales asociadas a partículas adheridas a la ventana de la sonda, eliminando las señales de tamaño que se repiten en el mismo punto durante varios muestreos. Este filtrado eliminaría la señal de un gran número de partículas primarias. Una discrepancia similar entre las distribuciones simuladas y medidas la observó Leba et al. (2010) en la simulación de DTC de agregados de gases hidratados.

En la figura 5.22 se recogen las DTC de un floculo virtual de MDF $307\ \mu\text{m}$ para distintas D_f

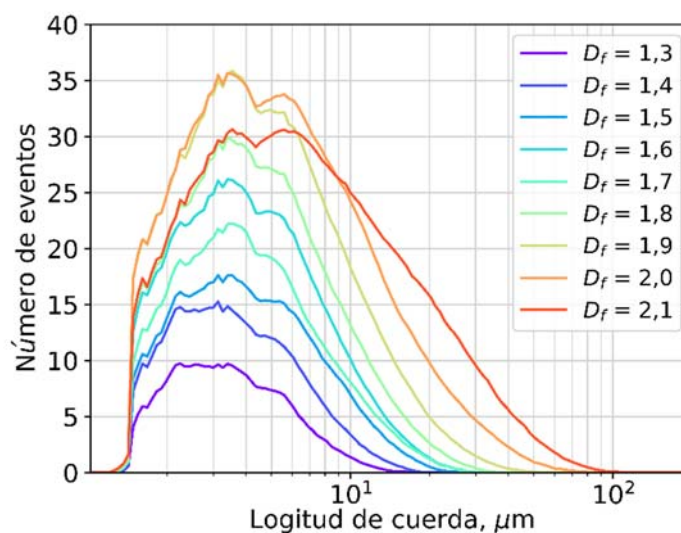


Figura 5.22. Longitudes de cuerda de flóculos de MDF $307\ \mu\text{m}$ y distintas dimensiones fractales

En la gráfica se observa que los flóculos con dimensiones fractales menores presentan valores máximos de tamaño detectado menores y menor número de eventos de detección que los de dimensiones fractales mayores. Como ocurriría en la realidad los flóculos de baja dimensión fractal presentan una mayor probabilidad de ser intersectados por el haz de láser a lo largo de segmentos más pequeños y de acuerdo a trayectorias cuya longitud caería por debajo del umbral de detección del equipo (Li and Wilkinson, 2005; Li et al., 2005; Yu et al., 2008).

5.4.2 Entrenamiento y selección del modelo de estimación de la D_f por medio de la DTC.

Las distribuciones de tamaño virtuales se emplearon en la selección de un modelo de regresión por bosques aleatorios. En la gráfica de la figura 5.23 se representa el valor de D_f estimada por el modelo frente al consignado en la construcción de los flóculos virtuales para cada uno de los grupos de datos de validación cruzada. Los valores medios del coeficiente de correlación $R = 0,98$ y RECM 0,003 respectivamente.

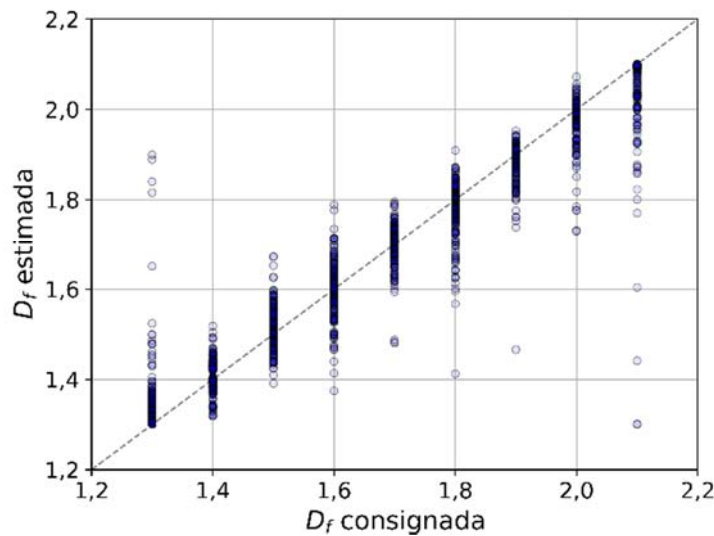


Figura 5.23. D_f estimada frente a consignada

Los hiper-parámetros del modelo final seleccionado se recogen en la tabla 5.6.

Tabla 5.6. Hiperparámetros seleccionados para el modelo de bosque aleatorio para la estimación de la dimensión fractal

Hiper-parámetros	Valor
Número de árboles	150
Máxima profundidad	102
Mínimo número de muestras permitido	71
Máximo número de variables considerado para la partición	178

5.4.3 Validación del modelo con datos reales.

El modelo seleccionado se validó con datos de DTC reales obtenidos de suspensiones de flóculos de microalgas con distintas concentraciones en las condiciones de operación consideradas en los ensayos previos con el propósito de comparar los resultados. La media de los valores de D_f estimados por el modelo de regresión durante un periodo de 150 segundos se recogen en la figura 5.24. En la gráfica también se incluye, a modo de referencia, el valor de

la D_f estimado por el método de comparación de las dimensiones fractales proyectadas descrito previamente.

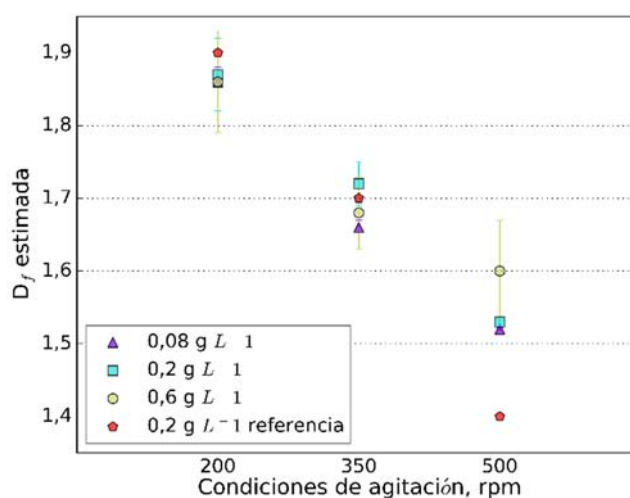


Figura 5.24. D_f estimada a partir de datos de DTC de suspensiones reales y la estimada por comparación para las velocidades de agitación consideradas.

Se aprecia en la gráfica que el modelo de regresión de bosques aleatorios es capaz de estimar D_f con valores muy cercanos a los resultantes del método de estimación por comparación para las condiciones de agitación de 200 y 350 rpm. En el caso de la agitación a 500 rpm, el modelo de predicción de la dimensión fractal tiende a sobreestimar el valor de este parámetro, especialmente en la máxima concentración de biomasa. Tal y como ocurre en el desarrollo del método de estimación de la dimensión fractal media de los flóculos a través de microscopía láser confocal, la dimensión fractal estimada es menor cuanto más intensas son las condiciones de agitación aplicadas en el ensayo. La coincidencia de los valores de estimación obtenidos con el método anterior y con este en condiciones de agitación intermedias sugiere que el método descrito en esta sección podría emplearse como sistema de estimación en tiempo real de la dimensión fractal media de flóculos de algas en suspensión. En vista de los resultados obtenidos, el presente método se ha empleado en la estimación de la dimensión fractal de agregados de caolín floculados por medio de celulosa nanocristalina hairy cationizada con resultados satisfactorios. La estimación directa de la dimensión fractal media de los flóculos podría servir, en conjunción con la medida de tamaños de cuerda de los mismos, para diseñar sistemas de control de la floculación que resulten en la producción de agregados de geometría óptima para etapas de separación subsiguientes, como se sugiere en de Oliveira et al. (2016).

6

CONCLUSIONES

En esta tesis doctoral se han desarrollado nuevas herramientas computacionales para la estimación de la concentración de biomasa de microalgas y para estimar la dimensión fractal media de suspensiones de agregados de microalgas. Estas herramientas están concebidas para posibilitar una monitorización más detallada y ágil de los procesos de cultivo y pre-concentración de suspensiones de microalgas con el objetivo último de facilitar la optimización de ambos procesos. El método de estimación de la concentración de biomasa desarrollado permitiría conocer el estado de crecimiento del cultivo con independencia de su estado de agregación, y los de estimación de la morfología de los agregados permitiría, en última instancia, optimizar la geometría de los agregados de microalgas por medio de las condiciones de agitación para mejorar el rendimiento de las etapas de concentración subsiguientes.

Los resultados alcanzados permiten obtener las siguientes conclusiones:

- Los modelos de regresión no lineal de aprendizaje automático son capaces de estimar la concentración de biomasa de microalgas a partir de datos de DTC con independencia del estado de agregación y concentración que presenta el cultivo. Corroboran esta conclusión los resultados obtenidos con los distintos tipos de modelos de regresión considerados y las dos cepas de microalgas estudiadas. Así, el modelo tipo perceptrón aplicado en los datos de DTC de suspensiones de *Chlamydomonas reihardtii* logró estimar la concentración de esta cepa con un error de predicción de $0.031 \text{ (g L}^{-1}\text{)}^2$ respecto a la concentración medida por gravimetría. Por otro lado, las estimaciones de concentración de biomasa realizadas con máquinas de vectores de soporte y bosques aleatorios a partir de datos de DTC de *Chlorella sorokiniana* presentaron ECM de predicción de $0,087 \text{ (g L}^{-1}\text{)}^2$ y $0,081 \text{ (g L}^{-1}\text{)}^2$ respectivamente.

- Dado el carácter prácticamente instantáneo de la medida de DTC por medio de la sonda láser por reflexión de haz enfocado los métodos de estimación de la concentración de biomasa permitirían hacer un seguimiento de la concentración de microalgas en su medio de cultivo, teniendo información, al mismo tiempo, de su estado de agregación. El empleo de esta técnica de estimación de la concentración permitiría evitar los posibles errores de medida en los que se incurre cuando se emplean técnicas de estimación ópticas en cultivos parcialmente agregados o con alta concentración de biomasa. Los modelos de estimación desarrollados se podrían entrenar con DTC de otros agregados de partículas distintas a las microalgas y detectables por la sonda láser para estimar la concentración de las suspensiones de tales flóculos.
- Se comprobó que la celulosa nanocristalina *hairy* cationizada es efectiva como agente de agregación de suspensiones de *Chlorella sorokiniana* en un amplio intervalo de dosis alrededor del punto isoelectrico. Se concluye, por tanto, que este nuevo tipo de nanopartícula puede emplearse como floculante en suspensiones de microalgas.
- Los resultados obtenidos en los ensayos de filtración confirman el papel determinante que el bionomio tamaño medio de flóculo y dimensión fractal media tiene en la filtrabilidad de suspensiones floculadas. Esta conclusión implica que la monitorización de dicho bionomio y la implementación de medidas de control encaminadas a garantizar que este se mantiene en los valores adecuados podrían contribuir a mejorar el rendimiento global del proceso de concentración de microalgas.
- En lo relativo a las herramientas informáticas diseñadas para estimar la dimensión fractal en 3D de los flóculos de microalgas, los resultados hacen patente que el sistema de crecimiento de los flóculos virtuales es altamente preciso en la construcción de agregados de MDF consignado en todas las morfologías consideradas. La correlación entre los tamaños consignados en la generación de los agregados y la media de los tamaños obtenidos en las nueve dimensiones fractales consideradas fue de $R^2 = 0,98$.
- El método de estimación de la D_f por comparación de dimensiones fractales proyectadas de imágenes microscópicas de flóculos reales con las de flóculos virtuales ofrece una alternativa a los métodos de estimación de la D_f convencionales para sistemas agregados compuestos por partículas primarias aproximadamente esféricas y con una dispersión de tamaños baja.
- Los resultados obtenidos en la estimación de la D_f consignada de flóculos virtuales por bosques aleatorios permiten concluir que este tipo de modelos de regresión es capaz de reconocer la dimensión fractal de flóculos simulados a partir de sus DTC ($R^2 = 0,96$).
- Cuando el modelo de estimación de la dimensión fractal por bosques aleatorios se aplica a datos de DTC reales, en vista de la similitud de los valores de D_f estimados por este y los obtenidos con el método de comparación de dimensiones fractales proyectadas, especialmente en condiciones de agitación suaves (desviación media = 0,15) se puede concluir que el modelo es una herramienta válida para la estimación de la morfología

media de los agregados de *C. sorokiniana*. Este modelo podría servir como herramienta de monitorización en el control del proceso de pre-concentración por floculación de agregados de la cepa considerada y de otras que presenten características similares.

- Los trabajos desarrollados para la estimación de la dimensión fractal tanto por comparación de las D_2 reales y virtuales, como el basado en correlacionar la DTC con D_6 , se pueden extender fácilmente a otros sistemas de agregados de partículas haciendo ligeras modificaciones a los algoritmos de crecimiento de flóculos virtuales para tener en cuenta la morfología de las partículas primarias que se consideren.

REFERENCIAS

- Abdullah, M.A., Shah, S.M.U., Shanab, S.M.M., Ali, H.E.A., (2017) Integrated Algal Bioprocess Engineering for Enhanced Productivity of Lipid, Carbohydrate and High-Value Bioactive Compounds. *Res. Rev. J. Microbiol. Biotechnol.* 6, 61-92.
- Acién, F., Molina, E., Reis, A., Torzillo, G., Zittelli, G., Sepúlveda, C., Masojídek, J., (2018) *Photobioreactors for the production of microalgae*. Microalgae-Based Biofuels and Bioproducts. Elsevier, pp. 1-44.
- Acién Fernández, F., Fernández Sevilla, J., Molina Grima, E., (2013) *Photobioreactors for the production of microalgae*. *Re/views in environmental science and bio/technology* 12, 131-151.
- Aguado, R., Moral, A., López, P., Mutjé, P., & Tijero, A. (2016). *Morphological analysis of pulps from orange tree trimmings and its relation to mechanical properties*. *Measurement*, 93, 319-326.
- Alonso Zorita, Á., (2014) Aplicación de simulación y redes neuronales a la industria del fibrocemento y del papel. Universidad Complutense de Madrid.
- Álvarez-Díaz, P., Ruiz, J., Arbib, Z., Barragán, J., Garrido-Pérez, M., Perales, J., (2017) *Freshwater microalgae selection for simultaneous wastewater nutrient removal and lipid production*. *Algal Research* 24, 477-485.
- Amjad, H., Khan, Z., Tarabara, V.V., (2015) *Fractal structure and permeability of membrane cake layers: effect of coagulation–flocculation and settling as pretreatment steps*. *Separation and Purification Technology* 143, 40-51.

- Anaya, P., Cárdenas, G., Lavayen, V., García, A., O'Dwyer, C., (2013) *Chitosan gel film bandages: Correlating structure, composition, and antimicrobial properties*. Journal of Applied Polymer Science 128, 3939-3948.
- Antunes, E., Garcia, F., Blanco, A., Negro, C., Rasteiro, M., (2015) *Evaluation of the flocculation and reflocculation performance of a system with calcium carbonate, cationic acrylamide co-polymers, and bentonite microparticles*. Industrial & Engineering Chemistry Research 54, 198-206.
- Arad, S.M., Ginzberg, A., Huleihel, M., (2006) *Antiviral activity of sulfated polysaccharides of marine red algae*. Biomaterials from Aquatic and Terrestrial Organizations. Enfield, NH: Science Publishers, 37-62.
- Ball, R., Weitz, D., Witten, T., Leyvraz, F., (1987) *Universal kinetics in reaction-limited aggregation*. Physical review letters 58, 274.
- Barros, A.I., Gonçalves, A.L., Simões, M., Pires, J.C., (2015) *Harvesting techniques applied to microalgae: a review*. Renewable and Sustainable Energy Reviews 41, 1489-1500.
- Basak, D., Pal, S., Patranabis, D.C., (2007) *Support vector regression*. Neural Information Processing-Letters and Reviews 11, 203-224.
- Beckmann, J., Lehr, F., Finazzi, G., Hankamer, B., Posten, C., Wobbe, L., Kruse, O., (2009) *Improvement of light to biomass conversion by de-regulation of light-harvesting protein translation in Chlamydomonas reinhardtii*. Journal of Biotechnology 142, 70-77.
- Bernaerts, T.M., Kyomugasho, C., Van Looveren, N., Gheysen, L., Foubert, I., Hendrickx, M.E., Van Loey, A.M., (2018) *Molecular and rheological characterization of different cell wall fractions of Porphyridium cruentum*. Carbohydrate polymers 195, 542-550.
- Berntson, G., Stoll, P., (1997) *Correcting for finite spatial scales of self-similarity when calculating fractal dimensions of real-world structures*. Proceedings of the Royal Society of London B: Biological Sciences 264, 1531-1537.
- Blanco, A., Fuente, E., Negro, C., Tijero, J., (2002) *Focused beam reflectant measurement as a tool to measure flocculation*. TAPPI J 100, 14-20.
- Bisoi, A.K., Mishra, J., (2001) *On calculation of fractal dimension of images*. Pattern Recognition Letters 22, 631-637.
- Bohachevsky, I.O., Johnson, M.E., Stein, M.L., (1986) *Generalized simulated annealing for function optimization*. Technometrics 28, 209-217.

- Borowitzka, M., (2018) *Commercial-Scale Production of Microalgae for Bioproducts*. Blue Biotechnology: Production and Use of Marine Molecules 1, 33-65.
- Borowitzka, M.A., Moheimani, N.R., (2013) *Open pond culture systems*. Algae for biofuels and energy. Springer, pp. 133-152.
- Breiman, L., (2001) *Random forests*. Machine learning 45, 5-32.
- Brennan, L., Owende, P., (2010) *Biofuels from microalgae—a review of technologies for production, processing, and extractions of biofuels and co-products*. Renewable and sustainable energy reviews 14, 557-577.
- Campano, C., Lopez-Exposito, P., Blanco, A., Negro, C., & van de Ven, T. G. (2019). Hairy cationic nanocrystalline cellulose as a novel flocculant of clay. Journal of Colloid and Interface Science.
- Cardozo, K.H., Guaratini, T., Barros, M.P., Falcão, V.R., Tonon, A.P., Lopes, N.P., Campos, S., Torres, M.A., Souza, A.O., Colepicolo, P., (2007) *Metabolites from algae with economical impact*. Comparative Biochemistry and Physiology Part C: Toxicology & Pharmacology 146, 60-78.
- Carman, P.C., (1937) *Fluid flow through granular beds*. Trans. Inst. Chem. Eng. 15, 150-166.
- Cazzaniga, S., Dall'Osto, L., Szaub, J., Scibilia, L., Ballottari, M., Purton, S., Bassi, R., (2014) *Domestication of the green alga Chlorella sorokiniana: reduction of antenna size improves light-use efficiency in a photobioreactor*. Biotechnology for biofuels 7, 157.
- Cosgun, E., Limdi, N.A., Duarte, C.W., (2011) *High-dimensional pharmacogenetic prediction of a continuous trait using machine learning techniques with application to warfarin dose prediction in African Americans*. Bioinformatics 27, 1384-1389.
- Chakraborti, R.K., Gardner, K.H., Atkinson, J.F., Van Benschoten, J.E., (2003) *Changes in fractal dimension during aggregation*. Water Research 37, 873-883.
- Chatsungnoen, T., Chisti, Y., (2016) *Harvesting microalgae by flocculation–sedimentation*. Algal Research 13, 271-283.
- Chen, Y.-H., Chang, G.-K., Kuo, S.-M., Huang, S.-Y., Hu, I.-C., Lo, Y.-L., Shih, S.-R., (2016) *Well-tolerated Spirulina extract inhibits influenza virus replication and reduces virus-induced mortality*. Scientific Reports 6, 24253.
- Chib, S., Greenberg, E., (1995) *Understanding the metropolis-hastings algorithm*. The american statistician 49, 327-335.

- Chiranjeevi, P., Mohan, S.V., (2016) *Critical parametric influence on microalgae cultivation towards maximizing biomass growth with simultaneous lipid productivity*. Renewable Energy 98, 64-71.
- Chisti, Y., (2013) *Raceways-based production of algal crude oil*. Green 3, 195-216.
- Cho, M.-H., Lee, C.-H., Lee, S., (2005) *Influence of floc structure on membrane permeability in the coagulation-MF process*. Water Science and Technology 51, 143-150.
- Choy, S.Y., Prasad, K.M.N., Wu, T.Y., Raghunandan, M.E., Phang, S.-M., Juan, J.C., Ramanan, R.N., (2018) *Separation of Chlorella biomass from culture medium by flocculation with rice starch*. Algal Research.
- Christenson, L., Sims, R., (2011) *Production and harvesting of microalgae for wastewater treatment, biofuels, and bioproducts*. Biotechnology advances 29, 686-702.
- Dassey, A.J., Theegala, C.S., (2013) *Harvesting economics and strategies using centrifugation for cost effective separation of microalgae cells for biodiesel applications*. Bioresource technology 128, 241-245.
- De Clercq, B., Lant, P.A., Vanrolleghem, P.A., (2004) *Focused beam reflectance technique for in situ particle sizing in wastewater treatment settling tanks*. Journal of Chemical Technology & Biotechnology: International Research in Process, Environmental & Clean Technology 79, 610-618.
- de Jesus Raposo, M.F., de Morais, A.M.M.B., (2015) *Microalgae for the prevention of cardiovascular disease and stroke*. Life sciences 125, 32-41.
- de Oliveira, A.L., Moreno, P., da Silva, P.A.G., De Julio, M., Moruzzi, R.B., (2016) *Effects of the fractal structure and size distribution of flocs on the removal of particulate matter*. Desalination and Water Treatment 57, 16721-16732.
- Devaraj, S., Jialal, I., Vega-López, S., (2004) *Plant sterol-fortified orange juice effectively lowers cholesterol levels in mildly hypercholesterolemic healthy individuals*. Arteriosclerosis, thrombosis, and vascular biology 24, e25-e28.
- Devos, M., Poisson, L., Ergon, F., Pencreac'h, G., (2006) *Enzymatic hydrolysis of phospholipids from Isochrysis galbana for docosahexaenoic acid enrichment*. Enzyme and microbial technology 39, 548-554.
- Dong, C., Chen, W., Liu, C., (2014) *Flocculation of algal cells by amphoteric chitosan-based flocculant*. Bioresource technology 170, 239-247.

- Doube, M., Kłosowski, M.M., Arganda-Carreras, I., Cordelières, F.P., Dougherty, R.P., Jackson, J.S., Schmid, B., Hutchinson, J.R., Shefelbine, S.J., (2010) *BoneJ: free and extensible bone image analysis in ImageJ*. Bone 47, 1076-1079.
- Dowsland, K.A., Thompson, J.M., (2012) *Simulated annealing*. Handbook of natural computing. Springer, pp. 1623-1655.
- Drucker, H., Burges, C.J., Kaufman, L., Smola, A.J., Vapnik, V., (1997) Support vector regression machines. Advances in neural information processing systems, pp. 155-161.
- Druine, F., Verney, R., Deloffre, J., Lemoine, J.-P., Chapalain, M., Landemaine, V., Lafite, R., (2018) *In situ high frequency long term measurements of suspended sediment concentration in turbid estuarine system (Seine Estuary, France): Optical turbidity sensors response to suspended sediment characteristics*. Marine Geology 400, 24-37.
- Eldridge, R., Hill, D., Gladman, B., (2012) *A comparative study of the coagulation behaviour of marine microalgae*. Journal of Applied Phycology 24, 1667-1679.
- Enzing, C., Ploeg, M., Barbosa, M., Sijtsma, L., (2014) *Microalgae-based products for the food and feed sector: an outlook for Europe*. Vigani, M., Parisi, C., Rodríguez Cerezo, E., Eds.
- Fasaei, F., Bitter, J., Slegers, P., van Boxtel, A., (2018) *Techno-economic evaluation of microalgae harvesting and dewatering systems*. Algal research 31, 347-362.
- Ferron, R.D., Shah, S., Fuente, E., Negro, C., (2013) *Aggregation and breakage kinetics of fresh cement paste*. Cement and Concrete Research 50, 1-10.
- Fields, F.J., Ostrand, J.T., Mayfield, S.P., (2018) *Fed-batch mixotrophic cultivation of Chlamydomonas reinhardtii for high-density cultures*. Algal Research 33, 109-117.
- Filippov, A., Zurita, M., Rosner, D., (2000) *Fractal-like aggregates: relation between morphology and physical properties*. Journal of Colloid and Interface Science 229, 261-273.
- Forrest, S., Witten Jr, T., (1979) *Long-range correlations in smoke-particle aggregates*. Journal of Physics A: Mathematical and General 12, L109.
- Fuente, E., Blanco, A., Negro, C., San Pio, I., Tijero, J., (2003) *Monitoring flocculation of fillers in papermaking*. Paper technology(1989) 44, 41-50.
- Gerde, J.A., Yao, L., Lio, J., Wen, Z., Wang, T., (2014) *Microalgae flocculation: impact of flocculant type, algae species and cell concentration*. Algal Research 3, 30-35.

- Gorman, D.S., Levine, R., (1965) *Cytochrome f and plastocyanin: their sequence in the photosynthetic electron transport chain of Chlamydomonas reinhardtii*. Proceedings of the National Academy of Sciences 54, 1665-1669.
- Gouveia, L., Marques, A.E., Sousa, J.M., Moura, P., Bandarra, N.M., (2010) *Microalgae-source of natural bioactive molecules as functional ingredients*. Food Sci Technol Bull Funct Foods 7, 21.
- Grima, E.M., Belarbi, E.-H., Fernández, F.A., Medina, A.R., Chisti, Y., (2003) *Recovery of microalgal biomass and metabolites: process options and economics*. Biotechnology advances 20, 491-515.
- Guccione, A., Biondi, N., Sampietro, G., Rodolfi, L., Bassi, N., Tredici, M.R., (2014) *Chlorella for protein and biofuels: from strain selection to outdoor cultivation in a Green Wall Panel photobioreactor*. Biotechnology for biofuels 7, 84.
- Guedes, A.C., Amaro, H.M., Malcata, F.X., (2011) *Microalgae as sources of carotenoids*. Marine drugs 9, 625-644.
- Hamed, I., Özogul, F., Özogul, Y., Regenstein, J.M., (2015) *Marine bioactive compounds and their health benefits: a review*. Comprehensive reviews in food science and food safety 14, 446-465.
- Hamed, I., Özogul, F., Regenstein, J.M., (2016) *Industrial applications of crustacean by-products (chitin, chitosan, and chitooligosaccharides): A review*. Trends in Food Science & Technology 48, 40-50.
- Hanotu, J., Bandulasena, H.H., Zimmerman, W.B., (2012) *Microflotation performance for algal separation*. Biotechnology and bioengineering 109, 1663-1673.
- Heinrich, J., Ulrich, J., (2012) *Application of Laser-Backscattering Instruments for In Situ Monitoring of Crystallization Processes—A Review*. Chemical Engineering & Technology 35, 967-979.
- Höpfner, T., Bluma, A., Rudolph, G., Lindner, P., Scheper, T., (2010) *A review of non-invasive optical-based image analysis systems for continuous bioprocess monitoring*. Bioprocess and biosystems engineering 33, 247-256.
- Illman, A., Scragg, A., Shales, S., (2000) *Increase in Chlorella strains calorific values when grown in low nitrogen medium*. Enzyme and microbial technology 27, 631-635.
- Jarabo, R., Fuente, E., Moral, A., Blanco, Á., Izquierdo, L., Negro, C., (2010) *Effect of sepiolite on the flocculation of suspensions of fibre-reinforced cement*. Cement and Concrete Research 40, 1524-1530.

- Jarabo, R., Monte, M.C., Blanco, A., Negro, C., Tijero, J., (2012) *Characterisation of agricultural residues used as a source of fibres for fibre-cement production*. Industrial Crops and Products 36, 14-21.
- Jeffers, P., Raposo, S., Lima-Costa, M.-E., Connolly, P., Glennon, B., Kieran, P.M., (2003) *Focussed beam reflectance measurement (FBRM) monitoring of particle size and morphology in suspension cultures of Morinda citrifolia and Centaurea calcitrapa*. Biotechnology letters 25, 2023-2028.
- Jiang, Q., Logan, B.E., (1991) *Fractal dimensions of aggregates determined from steady-state size distributions*. Environmental Science & Technology 25, 2031-2038.
- Jiang, Q., Logan, B.E., (1996) *Fractal dimensions of aggregates from shear devices*. Journal-American Water Works Association 88, 100-113.
- Johnson, C.P., Li, X., Logan, B.E., (1996) *Settling velocities of fractal aggregates*. Environmental science & technology 30, 1911-1918.
- Jullien, R., Thouy, R., Ehrburger-Dolle, F., (1994) *Numerical investigation of two-dimensional projections of random fractal aggregates*. Physical Review E 50, 3878.
- Kara, S., Mueller, J.J., Liese, A., (2011) *Online analysis methods for monitoring of bioprocesses*. Chimica Oggi-chemistry today 29, 38-41.
- Kovalsky, P., Bushell, G., (2005) *In situ measurement of fractal dimension using focussed beam reflectance measurement*. Chemical Engineering Journal 111, 181-188.
- Kozeny, J., (1927) *Über kapillare leitung der wasser in boden*. Royal Academy of Science, Vienna, Proc. Class I 136, 271-306.
- Kyzas, G.Z., Matis, K.A., (2016) *Electroflotation process: a review*. Journal of Molecular Liquids 220, 657-664.
- Lambrou, T.P., Anastasiou, C.C., Panayiotou, C.G., (2009) *A nephelometric turbidity system for monitoring residential drinking water quality*. International Conference on Sensor Applications, Experimentation and Logistics. Springer, pp. 43-55.
- Leba, H., Cameirao, A., Herri, J.-M., Darbouret, M., Peytavy, J.-L., Glénat, P., (2010) *Chord length distributions measurements during crystallization and agglomeration of gas hydrate in a water-in-oil emulsion: Simulation and experimentation*. Chemical Engineering Science 65, 1185-1200.

- Lee, A.K., Lewis, D.M., Ashman, P.J., (2009) *Microbial flocculation, a potentially low-cost harvesting technique for marine microalgae for the production of biodiesel*. Journal of Applied Phycology 21, 559-567.
- Lee, A.K., Lewis, D.M., Ashman, P.J., (2013) *Harvesting of marine microalgae by electroflocculation: the energetics, plant design, and economics*. Applied Energy 108, 45-53.
- Lee, S., Fane, A., Amal, R., Waite, T., (2003) *The effect of floc size and structure on specific cake resistance and compressibility in dead-end microfiltration*. Separation Science and Technology 38, 869-887.
- Li, J., Sun, C., Du, Q., (2006) A new box-counting method for estimation of image fractal dimension. Image Processing, 2006 IEEE International Conference on. IEEE, pp. 3029-3032.
- LI, L., GAO, Y.-t., DAI, Y., YANG, Y.-l., WANG, X.-m., (2007a) *Scavenging Effects of Spirulina and Polysaccharides Spirulina Platensis on Active Oxygens and Its Antioxidation in vitro [J]*. Chemistry & Bioengineering 3, 018.
- Li, M., Wilkinson, D., (2005) *Determination of non-spherical particle size distribution from chord length measurements. Part 1: Theoretical analysis*. chemical engineering science 60, 3251-3265.
- Li, M., Wilkinson, D., Patchigolla, K., (2005) *Determination of non-spherical particle size distribution from chord length measurements. Part 2: Experimental validation*. Chemical Engineering Science 60, 4992-5003.
- Li, T., Zhu, Z., Wang, D., Yao, C., Tang, H., (2007b) *The strength and fractal dimension characteristics of alum-kaolin flocs*. International Journal of Mineral Processing 82, 23-29.
- Li, Y., Horsman, M., Wu, N., Lan, C.Q., Dubois-Calero, N., (2008) *Biofuels from microalgae*. Biotechnology progress 24, 815-820.
- Liao, J.Y., Selomulya, C., Bushell, G., Bickert, G., Amal, R., (2005) *On different approaches to estimate the mass fractal dimension of coal aggregates*. Particle & Particle Systems Characterization 22, 299-309.
- Liao, Q., Chang, J.-s., Herrmann, C., Xia, A., (2018) *Bioreactors for Microbial Biomass and Energy Conversion*. Springer.
- Liu, L., Pohnert, G., Wei, D., (2016) *Extracellular metabolites from industrial microalgae and their biotechnological potential*. Marine drugs 14, 191.

- Logan, B.E., Kilps, J.R., (1995) *Fractal dimensions of aggregates formed in different fluid mechanical environments*. Water Research 29, 443-453.
- Ma, Z., Merkus, H.G., de Smet, J.G., Heffels, C., Scarlett, B., (2000) *New developments in particle characterization by laser diffraction: size and shape*. Powder Technology 111, 66-78.
- Maeda, Y., Yoshino, T., Matsunaga, T., Matsumoto, M., Tanaka, T., (2018) *Marine microalgae for production of biofuels and chemicals*. Current opinion in biotechnology 50, 111-120.
- Mandelbrot, B., (1975) *Les Objects Fractals: Forme, Hasard et Dimension*. flammarion. Paris & Montreal.
- Mathias, K.E., Whitley, L.D., (1994) Transforming the search space with gray coding. Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on. IEEE, pp. 513-518.
- Mathimani, T., Mallick, N., (2018) *A comprehensive review on harvesting of microalgae for biodiesel—Key challenges and future directions*. Renewable and Sustainable Energy Reviews 91, 1103-1120.
- McCulloch, W.S., Pitts, W., (1943) *A logical calculus of the ideas immanent in nervous activity*. The bulletin of mathematical biophysics 5, 115-133.
- McDonald, K.A., Jackman, A.P., Hurst, S., (2001) *Characterization of plant suspension cultures using the focused beam reflectance technique*. Biotechnology letters 23, 317-324.
- Meakin, P., (1983) *Formation of fractal clusters and networks by irreversible diffusion-limited aggregation*. Physical Review Letters 51, 1119.
- Mersmann, A., (2001) *Crystallization technology handbook*. CRC Press.
- Millach, L., Obiol, A., Solé, A., Esteve, I., (2017) *A novel method to analyse in vivo the physiological state and cell viability of phototrophic microorganisms by confocal laser scanning microscopy using a dual laser*. Journal of microscopy 268, 53-65.
- Milledge, J.J., Heaven, S., (2013) *A review of the harvesting of micro-algae for biofuel production*. Reviews in Environmental Science and Bio/Technology 12, 165-178.
- Miranda, R., Blanco, A., Fuente, E.d.l., Negro, C., (2008) *Separation of contaminants from deinking process water by dissolved air flotation: effect of flocculant charge density*. Separation Science and Technology 43, 3732-3754.

- Mobin, S., Alam, F., (2017) *Some promising microalgal species for commercial applications: A review*. Energy Procedia 110, 510-517.
- Moruzzi, R.B., de Oliveira, A.L., da Conceição, F.T., Gregory, J., Campos, L.C., (2017) *Fractal dimension of large aggregates under different flocculation conditions*. Science of the Total Environment 609, 807-814.
- Murthy, K.C., Vanitha, A., Rajesha, J., Swamy, M.M., Sowmya, P., Ravishankar, G.A., (2005) *In vivo antioxidant activity of carotenoids from Dunaliella salina—a green microalga*. Life Sciences 76, 1381-1390.
- Negro, C., Blanco, A., Fuente, E., Sánchez, L.M., Tijero, J., (2005) *Influence of flocculant molecular weight and anionic charge on flocculation behaviour and on the manufacture of fibre cement composites by the Hatschek process*. Cement and Concrete Research 35, 2095-2103.
- Negro, C., Sánchez, L.M., Fuente, E., Blanco, Á., Tijero, J., (2006) *Polyacrylamide induced flocculation of a cement suspension*. Chemical Engineering Science 61, 2522-2532.
- Nicu, R., Bobu, E., Miranda, R., Blanco, A., (2012) *Flocculation efficiency of chitosan for papermaking applications*. BioResources 8, 768-784.
- Nishijima, K., Kanasugi, A., Ando, K., (2014) *Accuracy improvement of genetic algorithm for obtaining floating-point solution*. Artificial Life and Robotics 19, 328-332.
- Noone, S., Ratcliff, K., Davis, R., Subramanian, V., Meuser, J., Posewitz, M.C., King, P.W., Ghirardi, M.L., (2017) *Expression of a clostridial [FeFe]-hydrogenase in Chlamydomonas reinhardtii prolongs photo-production of hydrogen from water splitting*. Algal research 22, 116-121.
- Norton, T.A., Melkonian, M., Andersen, R.A., (1996) *Algal biodiversity*. Phycologia 35, 308-326.
- Orhan, O., Haffner-Staton, E., La Rocca, A., Fay, M., (2016) *Characterisation of flame-generated soot and soot-in-oil using electron tomography volume reconstructions and comparison with traditional 2D-TEM measurements*. Tribology International 104, 272-284.
- Paddock, S.W., (2000) *Principles and practices of laser scanning confocal microscopy*. Molecular biotechnology 16, 127-149.
- Park, P.-K., Lee, C.-H., Lee, S., (2006) *Permeability of collapsed cakes formed by deposition of fractal aggregates upon membrane filtration*. Environmental science & technology 40, 2699-2705.

- Pashkow, F.J., Watumull, D.G., Campbell, C.L., (2008) *Astaxanthin: a novel potential treatment for oxidative stress and inflammation in cardiovascular disease*. The American journal of cardiology 101, S58-S68.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., (2011) *Scikit-learn: Machine learning in Python*. Journal of Machine Learning Research 12, 2825-2830.
- Prance, G.T., Beentje, H., Dransfield, J., Johns, R., (2000) *The tropical flora remains undercollected*. Annals of the Missouri Botanical Garden, 67-71.
- Preikschat, F.K., Preikschat, E., (1991) Apparatus and method for particle analysis. Google Patents.
- Priyadarshani, I., Rath, B., (2012) *Commercial and industrial applications of micro algae—A review*. J algal biomass utln 3, 89-100.
- Pulz, O., Gross, W., (2004) *Valuable products from biotechnology of microalgae*. Applied microbiology and biotechnology 65, 635-648.
- Raghuvanshi, S., Bhakar, V., Chava, R., Sangwan, K., (2018) *Comparative Study Using Life Cycle Approach for the Biodiesel Production from Microalgae Grown in Wastewater and Fresh Water*. Procedia CIRP 69, 568-572.
- Raj, P., Batchelor, W., Blanco, A., de la Fuente, E., Negro, C., Garnier, G., (2016) *Effect of polyelectrolyte morphology and adsorption on the mechanism of nanocellulose flocculation*. Journal of colloid and interface science 481, 158-167.
- Raj, P., Blanco, A., de la Fuente, E., Batchelor, W., Negro, C., Garnier, G., (2017) *Microfibrillated cellulose as a model for soft colloid flocculation with polyelectrolytes*. Colloids and Surfaces A: Physicochemical and Engineering Aspects 516, 325-335.
- Ramos-Martinez, E.M., Fimognari, L., Sakuragi, Y., (2017) *High-yield secretion of recombinant proteins from the microalga Chlamydomonas reinhardtii*. Plant biotechnology journal 15, 1214-1224.
- Ratha, S., Prasanna, R., (2012) *Bioprospecting microalgae as potential sources of “green energy”—challenges and perspectives*. Applied biochemistry and microbiology 48, 109-125.
- Riedmiller, M., (1994) *Advanced supervised learning in multi-layer perceptrons—from backpropagation to adaptive learning algorithms*. Computer standards and interfaces 16, 265-278.

- Rieseberg, L.H., (1997) *Hybrid origins of plant species*. Annual review of Ecology and Systematics 28, 359-389.
- Roselet, F., Vandamme, D., Roselet, M., Muylaert, K., Abreu, P.C., (2015) *Screening of commercial natural and synthetic cationic polymers for flocculation of freshwater and marine microalgae and effects of molecular weight and charge density*. Algal research 10, 183-188.
- Rosenblatt, F., (1958) *The perceptron: a probabilistic model for information storage and organization in the brain*. Psychological review 65, 386.
- Rubio, J., Souza, M., Smith, R., (2002) *Overview of flotation as a wastewater treatment technique*. Minerals engineering 15, 139-155.
- Ryckebosch, E., Bruneel, C., Muylaert, K., Foubert, I., (2012) *Microalgae as an alternative source of omega-3 long chain polyunsaturated fatty acids*. Lipid Technology 24, 128-130.
- Saarimaa, V., Sundberg, A., Holmbom, B., Blanco, A., Fuente, E., Negro, C., (2006) *Monitoring of dissolved air flotation by focused beam reflectance measurement*. Industrial & engineering chemistry research 45, 7256-7263.
- Sakamoto, H., TORADA, H., GOTO, K., NAKAMURA, Y., NAKANO, T., YAMAGUCHI, T., SATO, M., SAITO, T., TANIGUCHI, A., YOKOYAMA, T., (2003) *Biological activity of the polysaccharide produced by the marine phytoplankton Porphyridium sp. and additive effect of slag on the polysaccharide production*. Tetsu-to-Hagane 89, 475-481.
- Salama, E.-S., Kurade, M.B., Abou-Shanab, R.A., El-Dalatony, M.M., Yang, I.-S., Min, B., Jeon, B.-H., (2017) *Recent progress in microalgal biomass production coupled with wastewater treatment for biofuel generation*. Renewable and Sustainable Energy Reviews 79, 1189-1211.
- Salim, S., Bosma, R., Vermuë, M.H., Wijffels, R.H., (2011) *Harvesting of microalgae by bio-flocculation*. Journal of applied phycology 23, 849-855.
- Santley, R.S., Rose, E.K., (1979) *Diet, nutrition and population dynamics in the Basin of Mexico*. World Archaeology 11, 185-207.
- Schenk, P.M., Thomas-Hall, S.R., Stephens, E., Marx, U.C., Mussgnug, J.H., Posten, C., Kruse, O., Hankamer, B., (2008) *Second generation biofuels: high-efficiency microalgae for biodiesel production*. Bioenergy research 1, 20-43.
- Selim, S.Z., Alsultan, K., (1991) *A simulated annealing algorithm for the clustering problem*. Pattern recognition 24, 1003-1008.

- Show, K.-Y., Lee, D.-J., (2014) *Algal biomass harvesting*. Biofuels from Algae. Elsevier, pp. 85-110.
- Singh, G., Patidar, S., (2018) *Microalgae harvesting techniques: A review*. Journal of environmental management 217, 499-508.
- Sirin, S., (2013) Pre-concentration strategies for microalgae harvesting as biorefinery process chain. Universitat Rovira i Virgili.
- Skorupski, K., Mroczka, J., Wriedt, T., Riefler, N., (2014) *A fast and accurate implementation of tunable algorithms used for generation of fractal-like aggregate models*. Physica A: Statistical Mechanics and its Applications 404, 106-117.
- So, G.-B., So, H.-R., Jin, G.-G., (2017) *Enhancement of the Box-Counting Algorithm for fractal dimension estimation*. Pattern Recognition Letters 98, 53-58.
- Sonnleitner, B., Locher, G., Fiechter, A., (1992) *Biomass determination*. Journal of Biotechnology 25, 5-22.
- Sorokin, C., Krauss, R.W., (1959) *Maximum growth rates of Chlorella in steady-state and in synchronized cultures*. Proceedings of the National Academy of Sciences 45, 1740-1744.
- Strand, S.P., Danielsen, S., Christensen, B.E., Vårum, K.M., (2005) *Influence of chitosan structure on the formation and stability of DNA– chitosan polyelectrolyte complexes*. Biomacromolecules 6, 3357-3366.
- Suganya, T., Varman, M., Masjuki, H., Renganathan, S., (2016) *Macroalgae and microalgae as a potential source for commercial applications along with biofuels production: a biorefinery approach*. Renewable and Sustainable Energy Reviews 55, 909-941.
- Sun, Y., Wang, H., Guo, G., Pu, Y., Yan, B., (2014) *The isolation and antioxidant activity of polysaccharides from the marine microalgae Isochrysis galbana*. Carbohydrate polymers 113, 22-31.
- Tang, P., Greenwood, J., Raper, J.A., (2002) *A model to describe the settling behavior of fractal aggregates*. Journal of colloid and interface science 247, 210-219.
- Teng, D., (2012) *From chitin to chitosan*. Chitosan-based hydrogels, 1-33.
- Thissen, U., Pepers, M., Üstün, B., Melssen, W., Buydens, L., (2004) *Comparing support vector machines to PLS for spectral regression applications*. Chemometrics and Intelligent Laboratory Systems 73, 169-179.

- Udaiyappan, A.F.M., Hasan, H.A., Takriff, M.S., Abdullah, S.R.S., (2017) *A review of the potentials, challenges and current status of microalgae biomass applications in industrial wastewater treatment*. Journal of water process engineering 20, 8-21.
- Uduman, N., Qi, Y., Danquah, M.K., Forde, G.M., Hoadley, A., (2010a) *Dewatering of microalgal cultures: a major bottleneck to algae-based fuels*. Journal of renewable and sustainable energy 2, 012701.
- Uduman, N., Qi, Y., Danquah, M.K., Hoadley, A.F., (2010b) *Marine microalgae flocculation and focused beam reflectance measurement*. Chemical Engineering Journal 162, 935-940.
- Vahedi, A., Gorczyca, B., (2012) *Predicting the settling velocity of flocs formed in water treatment using multiple fractal dimensions*. Water research 46, 4188-4194.
- Van Haver, L., Nayar, S., (2017) *Polyelectrolyte flocculants in harvesting microalgal biomass for food and feed applications*. Algal Research 24, 167-180.
- Vandamme, D., Foubert, I., Fraeye, I., Meesschaert, B., Muylaert, K., (2012) *Flocculation of Chlorella vulgaris induced by high pH: role of magnesium and calcium and practical implications*. Bioresource technology 105, 114-119.
- Vandamme, D., Foubert, I., Muylaert, K., (2013) *Flocculation as a low-cost method for harvesting microalgae for bulk biomass production*. Trends in biotechnology 31, 233-239.
- Vandamme, D., Pontes, S.C.V., Goiris, K., Foubert, I., Pinoy, L.J.J., Muylaert, K., (2011) *Evaluation of electro-coagulation–flocculation for harvesting marine and freshwater microalgae*. Biotechnology and bioengineering 108, 2320-2329.
- Vapnik, V., (1995) *The Nature of Statistical Learning Theory*. Springer.
- Vigani, M., Parisi, C., Rodríguez-Cerezo, E., Barbosa, M.J., Sijtsma, L., Ploeg, M., Enzing, C., (2015) *Food and feed products from micro-algae: market opportunities and challenges for the EU*. Trends in Food Science & Technology 42, 81-92.
- Wan, C., Alam, M.A., Zhao, X.-Q., Zhang, X.-Y., Guo, S.-L., Ho, S.-H., Chang, J.-S., Bai, F.-W., (2015) *Current progress and future prospect of microalgal biomass harvest using various flocculation technologies*. Bioresource technology 184, 251-257.
- Wang, B., Lan, C.Q., Horsman, M., (2012) *Closed photobioreactors for production of microalgal biomasses*. Biotechnology advances 30, 904-912.
- Webb, R.H., (1996) *Confocal optical microscopy*. Reports on Progress in Physics 59, 427.

- Wentzel, M., Gorzawski, H., Naumann, K.-H., Saathoff, H., Weinbruch, S., (2003) *Transmission electron microscopical and aerosol dynamical characterization of soot aerosols*. Journal of aerosol science 34, 1347-1370.
- Wijesekara, I., Pangestuti, R., Kim, S.-K., (2011) *Biological activities and potential health benefits of sulfated polysaccharides derived from marine algae*. Carbohydrate polymers 84, 14-21.
- Wyatt, N.B., O'Hern, T.J., Shelden, B., Hughes, L.G., Mondy, L.A., (2013) *Size and structure of Chlorella zofingiensis/FeCl₃ flocs in a shear flow*. Biotechnology and bioengineering 110, 3156-3163.
- Yang, H., van de Ven, T.G., (2016) *Preparation of hairy cationic nanocrystalline cellulose*. Cellulose 23, 1791-1801.
- Yu, Z.Q., Chow, P.S., Tan, R.B., (2008) *Interpretation of focused beam reflectance measurement (FBRM) data via simulated crystallization*. Organic Process Research & Development 12, 646-654.
- Yuheng, W., Shengguang, Z., Na, L., Yixin, Y., (2011) *Influences of various aluminum coagulants on algae floc structure, strength and flotation effect*. Procedia Environmental Sciences 8, 75-80.
- Zakrajšek, N., Fuente, E., Blanco, A., Golob, J., (2009) *Influence of cationic starch adsorption on fiber flocculation*. Chemical Engineering & Technology: Industrial Chemistry-Plant Equipment-Process Engineering-Biotechnology 32, 1259-1265.
- Zhang, J., Sun, Z., Sun, P., Chen, T., Chen, F., (2014) *Microalgal carotenoids: Beneficial effects and potential in human health*. Food & function 5, 413-425.
- Zhang, J., Xia, W., Liu, P., Cheng, Q., Tahi, T., Gu, W., Li, B., (2010a) *Chitosan modification and pharmaceutical/biomedical applications*. Marine drugs 8, 1962-1987.
- Zhang, X., Hu, Q., Sommerfeld, M., Puruhito, E., Chen, Y., (2010b) *Harvesting algal biomass for biofuels using ultrafiltration membranes*. Bioresource technology 101, 5297-5304.
- Zheng, L., Watson, D., Johnston, B., Clark, R.L., Edrada-Ebel, R., Elseheri, W., (2009) *A chemometric study of chromatograms of tea extracts by correlation optimization warping in conjunction with PCA, support vector machines and random forest data modeling*. Analytica Chimica Acta 642, 257-265.
- Zhou, G., Moment, A., Cuff, J., Schafer, W., Orella, C., Sirota, E., Gong, X., Welch, C., (2014) *Process development and control with recent new FBRM, PVM, and IR*. Organic Process Research & Development 19, 227-235.

- Zhou, W., Min, M., Hu, B., Ma, X., Liu, Y., Wang, Q., Shi, J., Chen, P., Ruan, R., (2013) *Filamentous fungi assisted bio-flocculation: a novel alternative technique for harvesting heterotrophic and autotrophic microalgal cells*. Separation and Purification Technology 107, 158-165.
- Zhu, C., Lee, Y., (1997) *Determination of biomass dry weight of marine microalgae*. Journal of applied phycology 9, 189-194.

ANEXO I

Principales bloques de código implementados

GENERADOR JUEGOS DATOS 10-FOLDS

El programa que se presenta a continuación, prepara los datos para emplearlos en el entrenamiento y validación de modelos de aprendizaje automático aplicando el criterio de selección de validación cruzada (*k-folds cross validation* descrito previamente en la presente memoria). El resultado de este código es un archivo de datos con 10 grupos, cada uno de los cuales contiene un 70% de datos para entrenamiento y un 30% de datos de validación de manera que estos últimos son únicos en cada grupo. Este esquema de programa se emplea tanto en la preparación de los datos de FBRM para estimar la concentración como en los utilizados para estimar la dimensión fractal.

```
# -*- coding: utf-8 -*-
"""
Created on Thu Oct 29 14:59:52 2015
@author: plopez
"""

import numpy as np
import scipy.sparse as sp
import math
import operator
from operator import itemgetter, attrgetter, methodcaller
import matplotlib.pyplot as plt
import random
import pickle
import cPickle as pickle
import tarfile
import os
import csv
from spyderlib.utils.iofuncs import load_dictionary

path="D:\\python scripts\\Scripts ANTERIOR ORDENADOR\\Chlorella\\data 92"
filename = 'all data with biomass.spydata'p'
#filename = 'n_10_fold_data_float32.p'

tar = tarfile.open(os.path.join(filename, path), 'r')
tar.extractall()
extracted_files = tar.getnames()
for f in extracted_files:
    if f.endswith('.pickle'):
        with open(f, 'r') as fdesc:
            data = pickle.loads(fdesc.read())

data_dict = load_dictionary(filename)
xx=data_dict[0]['X']
yy=data_dict[0]['y']
xx=preprocessing.scale(X)
yy=y
lista= map(None,xx,yy)
lista_shuffled=random.sample(lista, len(lista))
test_elements= int(np.floor(len(lista)/10))
train_elements=int(len(lista)-test_elements)
folds=10 #int(len(lista_shuffled)/test_elements)
test_elements_last_fold= test_elements+np.mod(len(lista),10)
```

```

folds_data={}
test_ind=int(x=0)
train_ind=int(x=0)
en_points=range(len(lista_shuffled))
for i in range(folds):
#     shuffled=random.shuffle(lista)
    test_ind=i*test_elements
    if i < 8:

        test_lista= en_points[test_ind:test_ind+test_elements]
        train_lista=[x for x in en_points if x not in test_lista]
    elif i==8:
        test_lista= en_points[test_ind:test_ind+11]
        train_lista=[x for x in en_points if x not in test_lista]
    elif i==9:
        test_lista= en_points[test_ind:test_ind+11]
        train_lista=[x for x in en_points if x not in test_lista]

    folds_data[i]=([lista_shuffled[x] for x in test_lista],[lista_shuffled[x] for x in train_lista])

with open('10_fold_data_92_points_scaled_float32.p', 'wb') as fp:
    pickle.dump(folds_data, fp)

```

SVR SELECTOR

Este programa carga los datos de FBRM y biomasa seca preparados mediante el script anterior y lleva a cabo la búsqueda de los hiperparámetros que minimizan el error de predicción de un modelo de regresión por vectores de soporte (SVR) mediante un algoritmo de recocido simulado combinado con una validación cruzada.

```

# -*- coding: utf-8 -*-
"""
Created on Thu Jul 02 11:04:46 2015
@author: plopez
"""

import numpy as np
import numba
from numba import jit, autojit
import scipy.stats
from sklearn import svm
from sklearn.svm import SVR
import math
import operator
from sklearn import preprocessing
import matplotlib.pyplot as plt
import matplotlib.colors as cl
from sympy.combinatorics.graycode import GrayCode
import random
import pickle
import tarfile
import os

maxit=int(x=10000)
maxerr=0.08

```

```

Temp0=int(x=100)
alfa=np.float32(0.9988)#0.99965#0.9985#0.9985#0.985
npar=int(x=11)
clf=SVR()
clf_prov=SVR()
best_mod=SVR()

c1=np.float16(range(1,10))
c2=np.float16(range(1,10))*10
c3=np.float16(range(1,10))*100
c4=np.float16(range(1,38))*1000
cval=np.concatenate ((c1,c2,c3,c4), axis=0) #2^4
global cmap
cmap=zip (cval,list(GrayCode(6).generate_gray()))

g1=np.float16(range(1,10))*0.00001
g2=np.float16(range(1,10))*0.0001
g3=np.float16(range(1,10))*0.001
g4=np.float16(range(1,38))*0.01
gval=np.concatenate ((g1,g2,g3,g4), axis=0) #2^4
global gmap
gmap=zip (gval,list(GrayCode(6).generate_gray()))

e1=np.float16(range(1,10))*0.01
e2=np.float16(range(1,10))*0.1
e3=np.float16(range(1,10))
e4=np.float16(range(1,38))*10
evalue=np.concatenate ((e1,e2,e3,e4), axis=0) #2^4
global emap
emap=zip (evalue,list(GrayCode(6).generate_gray()))

path="D:\python scripts\Scripts ANTERIOR ORDENADOR\Chlorella\Datos 92"
filename= '10_fold_data_92_points_scaled_float32.p'
with open(os.path.join(path, filename), 'rb') as fp:
    data = pickle.load(fp)

def _fit(model,train_data_X,train_data_y,test_data_X,test_data_y):

    model.fit(train_data_X, train_data_y)
    yest=model.predict(test_data_X)
    sc=np.mean((test_data_y-yest)**2)
    return sc

def gen_pop():
    parametros_svr = []
    nbits=6
    fun=np.random.randint(1,4)

    if (fun== 1):
        parametros_svr.append('rbf')
    elif (fun== 2):
        parametros_svr.append('sigmoid')
    elif (fun== 3):
        parametros_svr.append('linear')
    pop=np.round(np.random.rand(1,nbits*3)).ravel()
    Cbits=''.join(str(np.int(e)) for e in pop[0:6])
    C=np.float32([x[:,0] for x in cmap if x[:,1] == Cbits ])

```

```
gbits=''.join(str(np.int(e)) for e in pop[6:12])
gamma=np.float32([x[:]][0] for x in gmap if x[:]][1] == gbits ])
ebits=''.join(str(np.int(e)) for e in pop[12:18])
epsilon=np.float32([x[:]][0] for x in emap if x[:]][1] == ebits ])
parametros_svr.append(C)
parametros_svr.append(gamma)
parametros_svr.append(epsilon)
```

```
return parametros_svr
```

```
def mutate(parametros_in):
```

```
    parametros = parametros_in
    elemento=np.random.randint(0,4)
    pos=np.random.randint(0,6)
    p=np.random.rand(1)[0]
    v_param=parametros[elemento]
    new_bits=[]
    kernels=['rbf', 'sigmoid', 'linear']
    if elemento==0:
        kernels.remove(parametros[0])
        if p>0.5:
            new_kernel=kernels[0]
        else:
            new_kernel=kernels[1]
        parametros[0]=new_kernel
    elif elemento>0:
        if elemento==1:
            cbits=[x[:]][1] for x in cmap if x[:]][0] == v_param]
            new_bits=list(cbits[0])
            if new_bits[pos]=='1':
                new_bits[pos]='0'
            else: new_bits[pos]='1'
            cbits=''.join(new_bits)
            parametros[1]=np.float32([x[:]][0] for x in cmap if x[:]][1] =
                = cbits ])

        elif elemento==2:
            gbits=[x[:]][1] for x in gmap if x[:]][0] == v_param]
            new_bits=list(gbits[0])
            if new_bits[pos]=='1':
                new_bits[pos]='0'
            else: new_bits[pos]='1'
            gbits=''.join(new_bits)
            parametros[2]=np.float32([x[:]][0] for x in gmap if x[:]][1] =
                = gbits ])

        elif elemento==3:
            ebits=[x[:]][1] for x in emap if x[:]][0] == v_param]
            new_bits=list(ebits[0])
            if new_bits[pos]=='1':
                new_bits[pos]='0'
            else: new_bits[pos]='1'
            ebits=''.join(new_bits)
            parametros[3]=np.float32([x[:]][0] for x in emap if x[:]][1] =
                = ebits ])

    return parametros
```

```

jfit=autofit(_fit)#####Define poblacion#####
R2=[]
iga=0
all_err= np.float32(np.empty(maxit))
parametros_svr=gen_pop()
error_interno=np.float32(np.empty((10,1)).ravel())
clf.set_params(kernel=parametros_svr[0], cache_size=200,
               C =parametros_svr[1], epsilon=parametros_svr[3],
               gamma = parametros_svr[2])

for ii in xrange(len(data)): #para cada uno de los folds
    test_data=[g for g in data[ii][0]]
    test_data_X=np.array([x[0] for x in test_data]) #7x200
    test_data_y=np.array([x[1] for x in test_data])
    # carga training data para el fold actual
    train_data=[g for g in data[ii][1]]
    train_data_X=np.array([x[0] for x in train_data])
    train_data_y=np.array([x[1] for x in train_data])
    error_interno[ii]=jfit(clf,train_data_X,train_data_y,test_data_X,
                          test_data_y)
error=np.mean(error_interno)
best_C= error
best_param=list(parametros_svr)
abs_best_C=error
abs_best_param=list(parametros_svr)
Temp=Temp0

for iga in xrange(maxit):

    all_err[iga]=best_C
    parametros_prov=list(best_param)
    mutate(parametros_prov) # muta un bit de los hiperparametros del
                           # modelo actual

    clf_prov.set_params(kernel=parametros_prov[0], cache_size=200,
                       C=parametros_prov[1], epsilon=parametros_prov[3],
                       gamma=parametros_prov[2])

    ### evalúa el modelo para los 10 folds y hace la media
    for ii in xrange(len(data)): #para cada uno de los folds

        test_data=[g for g in data[ii][0]]
        test_data_X=np.array([x[0] for x in test_data])
        test_data_y=np.array([x[1] for x in test_data])
        train_data=[g for g in data[ii][1]]
        train_data_X=np.array([x[0] for x in train_data])
        train_data_y=np.array([x[1] for x in train_data])
        error_interno[ii]=jfit(clf_prov,train_data_X,train_data_y,
                              test_data_X,test_data_y)
    err_prov=np.mean(error_interno)

    if err_prov < best_C:

        best_C=err_prov
        best_param=list(parametros_prov) #hace una copia independiente
        if best_C < abs_best_C:
            abs_best_C=best_C
            abs_best_param=best_param

```

```

elif err_prov > best_C:
    if Temp<1.e-6: Paccept=0
    else:
        Paccept=math.exp(-(err_prov- best_C)/Temp)
        P=np.random.rand(1)[0]
        if Paccept>P:
            best_C=err_prov
            best_param=list(parametros_prov)

if np.mod(iga,100)==0:

    print 'iga: ', iga, ' Temp: ', Temp , ' Current: ', err_prov, '
        Best: ', abs_best_C

if best_C < maxerr: break
Temp=Temp*alfa

ccorr=[]
pred_vect=np.zeros((1,0)).ravel()
y_vect=np.zeros((1,0)).ravel()
best_mod.set_params(kernel=abs_best_param[0], cache_size=200,
    C=abs_best_param[1], epsilon=abs_best_param[3],
    gamma=abs_best_param[2])

#grafica los resultados
colores=[]
colores=['b','g','y','c','r','salmon','violet','orange','yellowgreen',
    'purple']

for ii in xrange(len(data)): #para cada uno de los folds
    test_data=[g for g in data[ii][0]]
    test_data_X=np.array([x[0] for x in test_data])
    test_data_y=np.array([x[1] for x in test_data])

    train_data=[g for g in data[ii][1]]
    train_data_X=np.array([x[0] for x in train_data])
    train_data_y=np.array([x[1] for x in train_data])
    best_mod.fit(train_data_X, train_data_y)
    pred=best_mod.predict(test_data_X)
    plt.scatter(test_data_y,pred, color=colores[ii], marker='o')
    y_vect=np.concatenate((y_vect,test_data_y),)
    Rparcial=scipy.stats.linregress(test_data_y,pred)[2]
    ccorr.append(Rparcial)

plt.xlabel('Measured biomass concentration,
    g L-1 r"$^{-1}$",fontsize=13)
plt.ylabel('Estimated biomass concentration,
    g L-1 r"$^{-1}$",fontsize=13)
plt.plot(np.multiply(0.010,range(0,400,4)),np.multiply(0.010,
    range(0,400,4)),',',markersize=1, color='grey')
plt.axis([0, 2.5, 0, 2.5])
plt.grid(True)
plt.show()

```

RFR_SELECTOR

Este script carga los datos de FBRM y biomasa seca preparados mediante el programa *GENERADOR JUEGO DE DATOS 10-FOLDS* y lleva a cabo la búsqueda de los hiperparámetros que minimizan el error de predicción de un modelo de regresión por bosques aleatorios (RFR) mediante un algoritmo de recocido simulado combinado con una validación cruzada.

```
# -*- coding: utf-8 -*-
"""
Created on Thu Jul 02 11:04:46 2015
@author: plopez
"""

import numpy as np
import numba
from numba import jit, autojit
import scipy.stats
import math
from sklearn.ensemble import RandomForestRegressor
from sklearn import preprocessing
import matplotlib.pyplot as plt
import random
import pickle
import tarfile
import os
from sympy.combinatorics.graycode import GrayCode
global ne_map, nf_map, mins_map, maxd_map
clf=RandomForestRegressor()
clf_prov=RandomForestRegressor()
maxit=int(x=6000)
Temp0=int(x=100)
alfa=np.float16(0.9975)#0.9985#0.985
npar=int(x=11)
ne_val=np.float16(range(1,65))*15
ne_map=zip (ne_val,list(GrayCode(6).generate_gray()))
nf_val=[np.int(x) for x in np.round(np.float16(range(1,65))
    *3.076923077)]
nf_map=zip (nf_val,list(GrayCode(6).generate_gray()))
mins_val=np.round(np.float16(range(1,65))*3.076923077)
mins_map=zip (mins_val,list(GrayCode(6).generate_gray()))
maxd_val=np.round(np.float16(range(1,65))*3.076923077)
maxd_map=zip (mins_val,list(GrayCode(6).generate_gray()))
parametros_RF=list()
parametros_prov=list()
best_param=list()

def gen_pop_RF():

    parametros_rf = []
    nbits=6
    pop=np.round(np.random.rand(1,nbits*4)).ravel()
    ne_bits=''.join(str(np.int(e)) for e in pop[0:6])
    ne=np.float16([x[:][0] for x in ne_map if x[:][1] == ne_bits])
    ne=int(ne)
    nf_bits=''.join(str(np.int(e)) for e in pop[6:12])
```

```
nf=np.float16([x[:]][0] for x in nf_map if x[:] [1] == nf_bits ])
nf=int(nf)
mins_bits=''.join(str(np.int(e)) for e in pop[12:18])
mins=np.float16([x[:]][0] for x in mins_map if x[:] [1] == mins_bits ])
)
mins=int(mins)
maxd_bits=''.join(str(np.int(e)) for e in pop[18:24])
maxd=np.float16([x[:]][0] for x in maxd_map if x[:] [1] == maxd_bits ])
)
maxd=int(maxd)
parametros_rf.append(ne)
parametros_rf.append(nf)
parametros_rf.append(mins)
parametros_rf.append(maxd)

return parametros_rf
```

```
def mutate_RF(parametros_in):
```

```
    parametros = parametros_in
    elemento=np.random.randint(0,4)
    pos=np.random.randint(0,6)
    temp=0
    v_param=parametros[elemento]
    new_bits=[]

    if elemento==0:
        ne_bits=[x[:] [1] for x in ne_map if x[:] [0] == v_param]
        new_bits=list(ne_bits[0])
        if new_bits[pos]=='1':
            new_bits[pos]='0'
        else: new_bits[pos]='1'
        ne_bits=''.join(new_bits)
        temp=np.float16([x[:] [0] for x in ne_map if x[:] [1] == ne_bits ])
        )
        parametros[0]=int(temp[0])

    elif elemento==1:
        nf_bits=[x[:] [1] for x in nf_map if x[:] [0] == v_param]
        new_bits=list(nf_bits[0])
        if new_bits[pos]=='1':
            new_bits[pos]='0'
        else: new_bits[pos]='1'
        nf_bits=''.join(new_bits)
        temp=np.float16([x[:] [0] for x in nf_map if x[:] [1] == nf_bits ])
        )
        parametros[1]=int(temp[0])

    elif elemento==2:
        mins_bits=[x[:] [1] for x in mins_map if x[:] [0] == v_param]
        new_bits=list(mins_bits[0])
        if new_bits[pos]=='1':
            new_bits[pos]='0'
        else: new_bits[pos]='1'
        mins_bits=''.join(new_bits)
        temp=np.float16([x[:] [0] for x in mins_map if x[:] [1] ==
            mins_bits ])
        parametros[2]=int(temp[0])
```



```

elif elemento==3:
    maxd_bits=[x[:,1] for x in maxd_map if x[:,0] == v_param]
    new_bits=list(maxd_bits[0])
    if new_bits[pos]=='1':
        new_bits[pos]='0'
    else: new_bits[pos]='1'
    maxd_bits=''.join(new_bits)
    temp=np.float16([x[:,0] for x in maxd_map if x[:,1] ==
        maxd_bits ])
    parametros[3]=int(temp[0])

return parametros

path="D:\python scripts\Scripts ANTERIOR ORDENADOR\Chlorella\Datos 92"
filename= '10_fold_data_92_points_float32.p'
with open(os.path.join(path, filename), 'rb') as fp:
    data = pickle.load(fp)

def _fit(model,train_data_X,train_data_y,test_data_X,test_data_y):

    model.fit(train_data_X, train_data_y)
    yest=model.predict(test_data_X)
    return sc
jfit=autojit(_fit)#
ccorr=[]
iga=0
all_err= np.float16(np.empty(maxit))
error_interno=np.float16(np.empty((10,1)).ravel())

parametros_RF=gen_pop_RF()
clf.set_params(n_estimators=parametros_RF[0], max_features=
    parametros_RF[1], min_samples_split=parametros_RF[2],
    max_depth=parametros_RF[3])

for ii in xrange(10):
    test_data=[g for g in data[ii][0]]
    test_data_X=np.array([x[0] for x in test_data]) #7x200
    test_data_y=np.float32(np.array([x[1] for x in test_data]))
    train_data=[g for g in data[ii][1]]
    train_data_X=np.array([x[0] for x in train_data]) #70x200
    train_data_y=np.float32(np.array([x[1] for x in train_data]))
    error_interno[ii]=jfit(clf,train_data_X,train_data_y,test_data_X,
        test_data_y)
error=np.float16(np.mean(error_interno))
best_C= error
best_param=list(parametros_RF)
abs_best_C=error
abs_best_param=best_param
Temp=Temp0

for iga in xrange(maxit):
    all_err[iga]=best_C
    parametros_prov=list(best_param)
    mutate_RF(parametros_prov) #3mutate_RF(parametros_prov)
    clf_prov.set_params(n_estimators=parametros_prov[0],
        max_features=parametros_prov[1], min_samples_split=
        parametros_prov[2],max_depth=parametros_prov[3])

```

```

### evalua el modelo para los 10 folds y hace la media
for ii in xrange(len(data)): #para cada uno de los folds
    test_data=[g for g in data[ii][0]]
    test_data_X=np.array([x[0] for x in test_data]) #7x200
    test_data_y=np.float32(np.array([x[1] for x in test_data]))
    train_data=[g for g in data[ii][1]]
    train_data_X=np.array([x[0] for x in train_data]) #70x200
    train_data_y=np.float32(np.array([x[1] for x in train_data]))
    error_interno[ii]=jfit(clf_prov,train_data_X,train_data_y,
        test_data_X,test_data_y)
err_prov=np.mean(error_interno)

if err_prov < best_C:

    best_C=err_prov
    best_param=list(parametros_prov)

    if best_C < abs_best_C:

        abs_best_C=best_C
        abs_best_param=best_param
    else:
        Paccept=math.exp(-(err_prov- best_C)/Temp)
        P=np.random.rand(1)[0]
        if Paccept>P:
            best_C=err_prov
            best_param=list(parametros_prov)
if np.mod(iga, 1000)==0:
    print 'iga: ', iga, ' Temp: ', Temp , ' Current: ', err_prov, ' Best: ', best_C

    Temp=Temp*alfa
ccorr=[]
pred_vect=np.zeros((1,0)).ravel()

y_vect=np.zeros((1,0)).ravel()
clf.set_params(n_estimators=best_param[0], max_features=best_param[1], min_samples_split=best_param[2],max_depth=best_param[3])#oob_score=True)

colores=[]
colores=['b','g','y','c','r','salmon','violet','orange','yellowgreen','purple']
for ii in range(len(data)): #para cada uno de los folds
    test_data=[g for g in data[ii][0]]
    test_data_X=np.array([x[0] for x in test_data]) #7x200

    test_data_y=np.array([x[1] for x in test_data])

    train_data=[g for g in data[ii][1]]
    train_data_X=np.array([x[0] for x in train_data])
    train_data_y=np.array([x[1] for x in train_data])
    clf.fit(train_data_X, train_data_y)
    pred=clf.predict(test_data_X)
    plt.scatter(test_data_y,pred, color=colores[ii]) #marker='o',
    y_vect=np.concatenate((y_vect,test_data_y),)
    Rparcial=scipy.stats.linregress(test_data_y,pred)[2]
    ccorr.append(Rparcial)

```

```
plt.axis([0, 2.5, 0, 2.5])
plt.xlabel('Measured biomass concentration, g L-1', fontsize=13)
plt.ylabel('Estimated biomass concentration, g L-1', fontsize=13)
plt.plot(np.multiply(0.010, range(0, 400, 2)), np.multiply(0.010,
    range(0, 400, 2)), 'k--', Lw=1, markersize=1, color='grey')
plt.grid(True)
plt.show()
```

AGGREGATE

La clase *Aggregate* se creó para representar un flóculo de microalgas. Entre otros datos, una instancia de la clase *Aggregate* contiene información sobre la dimensión fractal del flóculo, las coordenadas espaciales de todas las células que lo forman, las coordenadas de su centro de masas, su radio de giro, el conjunto de células consideradas exteriores, la célula más alejada al centro de masas y el diámetro máximo de Feret.

```
# -*- coding: utf-8 -*-
"""
Created on Fri May 06 11:53:57 2016
@author: plopez
"""

import random, math
from random import jumpahead
import numpy as np
import numpy
import math
#from visual import *
from scipy.spatial import distance
from scipy import stats

class Aggregate:

    tolerance=0.00

    def PrintException(self):

        exc_type, exc_obj, tb = sys.exc_info()
        f = tb.tb_frame
        lineno = tb.tb_lineno
        filename = f.f_code.co_filename
        linecache.checkcache(filename)
        line = linecache.getline(filename, lineno, f.f_globals)
        print 'EXCEPTION IN ({}, LINE {} "{}"): {}'.format(filename,
            lineno, line.strip(), exc_obj)

    def __init__(self, k,d,Np, cells=None):

        self.cell_radius = np.float64(40.0)
        self.scale= 1.5/self.cell_radius
        if cells == None:
            self.kf=k
```

```

        self.df=d
        self.Np=Np
        self.cells=np.zeros((self.Np,3), dtype=np.float64)
        self.cells[0], self.cells[1]=
            np.array([0,0,0],dtype=np.float64),
            2.0*self.cell_radius * self.gen_angle()
        self.current_index_to_fill=2
        self.masses_centre=(self.cells[0]+self.cells[1])/2.0
        self.gyration_radius=0
        self.max_distance=self.cell_radius*2.0
        self.furthest_cell=self.cells[1]
        self.outer_cells=self.cells
        self.gamma_distance=np.float64(0.0)
        self.recalculate_gamma_distance()
        self.length=2*self.cell_radius*self.scale
    else:
        self.kf=k
        self.df=d
        self.Np=Np
        self.cells=cells
        self.current_index_to_fill=len(cells)
        self.masses_centre=None
        self.recalculate_masses_centre(2, None)
        self.gamma_distance=0
        self.recalculate_gamma_distance()
        self.gyration_radius=None
        self.recalculate_gyration_radius()
        self.max_distance=None #####
        self.outer_cells=None
        self.furthest_cell=None
        self.redefine_outer_cells()
        self.update_max_distance(1, None)
        self.length=0
        self.measure_length()

def update_floc(self):

    self.recalculate_masses_centre(2, None)

def gen_angle(self):

    x=random.uniform(-1,1)
    y=random.uniform(-1,1)
    z=random.uniform(-1,1)

    magnitud=math.sqrt(x**2+y**2+z**2)
    return np.array([x/magnitud,y/magnitud,z/magnitud],dtype=np.floa
t64)

def translate_cells(self, displacement_vector):

    aggregate_temp=np.empty((np.shape(self.cells)),dtype=
np.float64)
    outer_temp=np.empty(np.shape(self.outer_cells),dtype=
np.float64)
    for cell in xrange(self.Np):

```

```

        aggregate_temp[cell]= np.array(displacement_vector +
                                         self.cells[cell])
    for cell in xrange(len(self.outer_cells)):
        outer_temp[cell]= np.array(displacement_vector +
                                     self.outer_cells[cell])

    self.cells=aggregate_temp
    self.outer_cells=outer_temp
    self.recalculate_masses_centre(0, displacement_vector)

def rotation_matrix(self,angle, direction, point=None):

    sina = math.sin(angle)
    cosa = math.cos(angle)
    direction = self.unit_vector(direction[:3])
    # rotation matrix around unit vector
    R = numpy.diag([cosa, cosa, cosa])
    R += numpy.outer(direction, direction) * (1.0 - cosa)
    direction *= sina
    R += numpy.array([[ 0.0, - direction[2],  direction[1]],
                      [ direction[2], 0.0, -direction[0]],
                      [-direction[1], direction[0],  0.0]])

    M = numpy.identity(4)
    M[:3, :3] = R
    if point is not None:
        # rotacion excentrica respect al origen
        point = numpy.array(point[:3], dtype=numpy.float64,
                             copy=False)
        M[:3, 3] = point - numpy.dot(R, point)
    return M

def rotate_floc(self, rotation_matrix):

    rotated_cells=np.empty(shape(self.cells))
    m_centre=np.empty((1,3))
    m_centre=np.dot(rotation_matrix,self.masses_centre)
    for cell in xrange(len(self.cells)):
        rotated_cells[cell]=np.dot(rotation_matrix,
                                    self.cells[cell])
    self.cells=rotated_cells
    self.masses_centre=m_centre

def add_cells(self, option=None):

    if option==None:
        for cell in range(2,self.Np):
            selected_cell=np.empty((1,3))
            overlapped=True
            rotations=25
            while overlapped:
                count=0
                selected_cell= self.outer_cells[random.choice(
                    xrange(len(self.outer_cells)))]

                # selecciona una celula exterior aleatoria a
                #distancia gamma
                while count<rotations:

```

```

        count+=1
        potential_cell=self.get_potential_position(
            selected_cell) #rotacion
                               #implicita

        overlapped = self.is_overlapping(
            potential_cell) #comprueba si overlap
        if overlapped==False:
            count=25

        #recalcula
        self.recalculate_masses_centre(1, potential_cell)
        self.update_max_distance(0, potential_cell)
        #añade
        self.cells[self.current_index_to_fill]=potential_cell
        self.current_index_to_fill+=1
        self.recalculate_gyration_radius()
        self.recalculate_gamma_distance()
        self.redefine_outer_cells()
        self.measure_length()

    elif option==1:
        self.__init__ (self.kf,self.df,self.Np)
        self.add_cells()

def get_potential_position(self,cell):

    x0=self.masses_centre[0]
    y0=self.masses_centre[1]
    z0=self.masses_centre[2]
    x=cell[0]
    y=cell[1]
    z=cell[2]
    b=np.array([x,y,z])
    c=np.array([x0,y0,z0])
    C_B=np.float64(distance.euclidean(c,b))
    C_A=self.gamma_distance
    A_B=np.float64(2.0*self.cell_radius)
    alfa= np.float64(math.acos((C_A**2+C_B**2-
                                A_B**2)/(2*C_A*C_B)))
    C_Aprime=np.float64(C_B/math.sin(math.pi-alfa-
                                math.pi/2.0))
    B_Aprime=np.float64(C_Aprime*math.sin(alfa))
    v_CB= b-c
    A_Aprime=np.float64(C_Aprime - C_A)
    v_perp=self.perpendicular_vector(v_CB)
    rot_matrix=self.rotation_matrix(2.0*math.pi*random.uniform(0,1
                                ), v_CB,b-c)
    v_perp_rotated=np.dot(rot_matrix,np.concatenate((v_perp,[1]),
                                axis=0))
    v_perp_full=B_Aprime*v_perp_rotated[:3]
    v_Aprime=b+v_perp_full
    dir_A=v_Aprime-c

    vector_A0= C_A * dir_A/np.linalg.norm(dir_A)+c #prueba
    # comprobacion en 3D con visual python
    #vector_A0=c+C_A*dir_A/mag(dir_A)
    #vector_A0=C_A*dir_A/mag(dir_A)

```

```

        #vector_A=rotate(vector_A0, 2.0*math.pi*random.uniform(0,1), v
        _CB)
        #arrow(pos=c,axis=dir_A, length=self.gamma_distance, color=col
        or.green)
        #arrow(pos=b,axis=v_Aprime, length=mag(v_Aprime))
        #arrow(pos=c,axis=b-c, lenght=20, color=color.red)
        #alfa=rad*math.cos((sqdist+cb2-
            #(2.0*rad)**2)/(2.0*math.sqrt(sqdist*cb2)))
        #print "a0: %s" %(vector_A0)
        return np.array(vector_A0)    #genera vector unitario de otro
            dado

def unit_vector(self, v):

    return np.array([v[0]/np.linalg.norm(v), v[1]/np.linalg.norm(v),
v[2]/np.linalg.norm(v)
                    ]),dtype=np.float64)

def perpendicular_vector(self,v):    #genera vector unitario
    #perpendicular a otro dado

    if v[0] == v[1] == v[2] == 0:
        raise ValueError('zero-vector')
    if v[0] == 0:
        return np.array([1, 0, 0])
    if v[1] == 0:
        return np.array([0, 1, 0])
    if v[2] == 0:
        return np.array([0, 0, 1])
    perpendicular=np.array([1, 1, -
1 * (v[0] + v[1]) / v[2]],dtype=np.float64)
    return perpendicular/np.linalg.norm(perpendicular)

def r_new(a,b, radio):

    return vector(a+radio*2.0*b)

def is_overlapping (self, position): #comprueba si dos células
    #ocupan el mismo volumen

    chk=0
    for ii in xrange(self.current_index_to_fill):
        cell=self.cells[ii]
        #print "position %s" %(position)
        dist=math.sqrt((cell[0] -
position[0])**2 + (cell[1] - position[1])**2
+ (cell[2] - position[2])**2)
        if dist < 2.0*self.cell_radius:
            chk+=1
    if chk>0:
        return True
    else:
        return False

def recalculate_masses_centre(self, option , vector): #calcula
    #Nuevo centro masas floc cuando se traslada

```

```

if option == 0:
    self.masses_centre= vector + self.masses_centre
elif option ==1: #cuando se crea el agregado
    ncells=self.Np
    r00s=(ncells*self.masses_centre+vector)/(ncells+1)
    #print "centro " +np.str(r00s)
    self.masses_centre=r00s
    #print "c masas "+np.str(self.masses_centre)
elif option == 2: #cuando se crea el agregado mediante un
    #grupo de celulas dadas
    all_cells=np.empty((1,3))
    all_cells=self.cells.sum(axis=0)
    self.masses_centre=all_cells / self.Np

def recalculate_gyration_radius(self):    #recalcula radio giro

    all_rad=np.empty(len(self.cells))
    for ii in xrange(len(self.cells)):
        all_rad[ii]=distance.sqeuclidean(self.cells[ii],
            self.masses_centre)
    self.gyration_radius=math.sqrt(np.sum(all_rad)/
        len(self.cells))

def recalculate_gamma_distance(self):

    nump=self.current_index_to_fill
    rp=self.cell_radius
    k=self.kf
    d=self.df
    distancia2=np.float64(((nump*self.cell_radius)**2)*((nump/k)**(2
/d)))/(nump-1)-
    (nump*(self.cell_radius**2))/(nump-1)-nump*(self.cell_radius**2)*
        ((nump-1)/k)**(2/d))
    self.gamma_distance=np.float64(math.sqrt(distancia2))
    #print "gamma "+ np.str(self.check)+ ": "+np.str(self.gamma_di
#stance)

def redefine_outer_cells(self): #reclasifica celulas exteriores
    #del floculo

    valid_cells=np.empty((0,3))
    cm=self.masses_centre
    cell=np.empty((1,3))
    for ii in xrange(len(self.cells)):
        cell=self.cells[ii]
        dist=math.sqrt((cell[0]-
cm[0])**2 + (cell[1] - cm[1])**2 + (cell[2] - cm[2])**2)
        if dist > self.gamma_distance - 2.0*self.cell_radius :

            valid_cells=np.vstack((valid_cells,cell))
    #print "outer: "+np.str(valid_cells)
    self.outer_cells = valid_cells

```



```

def update_max_distance(self, option , potential_cell): #identific
    #a la celula mas lejana
    #del centro de masas
    if option==0:
        new_cell_dist = distance.euclidean(self.masses_centre,
            potential_cell)
        if new_cell_dist > self.max_distance:
            self.max_distance = new_cell_dist
            self.furthest_cell = potential_cell
    elif option == 1:
        for cell in xrange(len(self.outer_cells)):
            new_cell_dist = distance.euclidean(self.masses_centre,
                self.outer_cells[cell])#
            if new_cell_dist > self.max_distance:
                self.max_distance = new_cell_dist
                self.furthest_cell=self.outer_cells[cell]

def measure_length(self): #calcula el Feret maximo del
                                #floculo

    outer_cells=self.outer_cells
    furthest=self.furthest_cell

    distancias=np.empty((np.shape(outer_cells)[0],1))
    for ii in xrange(len(outer_cells)):
        distancias[ii]= distance.euclidean(furthest,
            outer_cells[ii])
#         print distancias
    longitud= np.max(distancias)
    longitud=(longitud+2*self.cell_radius)*self.scale #*3.5/10.0

    self.length=longitud
#         print "indice original: "+np.str(indice)

```

FLOCS MERGER

La clase *Flocs_merger* se implementó con el propósito de posibilitar la unión virtual de dos agregados de forma aleatoria y compatible con la integridad de ambos de tal manera que en el agregado resultante no haya dos células que compartan un mismo volumen, es decir, que se solapen, en el espacio. El método principal de la clase toma dos instancias de la clase *Aggregate* (2 flóculos virtuales), los traslada en el espacio, selecciona aleatoriamente una célula exterior en cada uno, determina preliminarmente si la unión a través de esas células es compatible y, de serlo, practica una serie de rotaciones de ambos flóculos en el espacio para conseguir el contacto de las células inicialmente seleccionadas.

```

# -*- coding: utf-8 -*-
"""
Created on Fri May 06 11:53:57 2016
@author: plopez
"""
from __future__ import division
import sys
import random, math
from random import jumpahead

```

```

from random import uniform, random
import numpy as np
from numpy import cross, eye, dot
import math
import numpy
import copy
from scipy.spatial import distance
from scipy.linalg import expm3, norm
from scipy import stats
from aggregate6 import Aggregate
from scipy import weave

class Flocs_merger:

#   cell_radius=10.0
#   tolerance=0.00

    def __init__(self, aggregate1, aggregate2):

        self.cell_radius= aggregate1.cell_radius
        self.aggregate1=aggregate1
        self.aggregate2=aggregate2
        self.aggregate_temp=np.empty(np.shape(aggregate2.cells))
        self.df=aggregate2.df
        self.kf=aggregate2.kf
        self.Np1=aggregate1.Np
        self.Np2=aggregate2.Np
        self.max_distance1= np.float64(aggregate1.max_distance)
        self.max_distance2= np.float64(aggregate2.max_distance)
        self._EPS = numpy.finfo(float).eps * 4.0
        self.gamma_distance=np.float64(math.sqrt( ( ((self.Np1+
            self.Np2)**2) *self.cell_radius**2 /
            (self.Np1*self.Np2))* (((self.Np1+self.Np2)/
            self.kf)**(2.0/self.df)) - (self.Np1+self.Np2) *
            self.aggregate1.gyration_radius**2 /
            self.Np2 - (self.Np1+self.Np2) *
            self.aggregate2.gyration_radius**2 / self.Np1 ))

    def gen_angle(self): #genera un vector director unitario aleatorio

        x=uniform(-1,1)#random.uniform(-1,1)
        y=uniform(-1,1)#random.uniform(-1,1)
        z=uniform(-1,1)#random.uniform(-1,1)
        magnitud=math.sqrt(x**2+y**2+z**2)
        return np.array([x/magnitud,y/magnitud,z/magnitud])

    def move_floc(self): #traslada el floculo 2 hacia el 1 en la
        #direccion aleatoria generada hasta posicionarlo a distancia
        #gamma

        attach_direction=self.aggregate1.masses_centre+
            self.gamma_distance*self.gen_angle()
        displacement=attach_direction - self.aggregate2.masses_centre
        self.aggregate2.translate_cells(displacement)

    def vector_norm(self, data, axis=None, out=None):

```

```

data = numpy.array(data, dtype=numpy.float64, copy=True)
if out is None:
    if data.ndim == 1:
        return math.sqrt(numpy.dot(data, data))
    data *= data
    out = numpy.atleast_1d(numpy.sum(data, axis=axis))
    numpy.sqrt(out, out)
    return out
else:
    data *= data
    numpy.sum(data, axis=axis, out=out)
    numpy.sqrt(out, out)

def rotation_matrix(self, angle, direction, point=None):
    #devuelve matriz de rotacion sobre un eje definido por un
    #punto y un vector de direccion

    sina = math.sin(angle)
    cosa = math.cos(angle)
    direction = self.unit_vector(direction[:3])
    # matriz rotacion alrededor de un vector unitario
    R = numpy.diag([cosa, cosa, cosa])
    R += numpy.outer(direction, direction) * (1.0 - cosa)
    direction *= sina
    R += numpy.array([[ 0.0, -direction[2], direction[1]],
                      [ direction[2], 0.0,          -
                        direction[0]], [-direction[1],
                        direction[0], 0.0]])
    M = numpy.identity(4)
    M[:3, :3] = R
    if point is not None:
        # rotacion excentrica
        point = numpy.array(point[:3], dtype=numpy.float64,
                             copy=False)
        M[:3, 3] = point - numpy.dot(R, point)
    return M

def create_intersection_list (self, aggregate1, aggregate2):
    #genera lista de celulas exteriores potencialmente
    intersectables

    inner_radius= self.gamma_distance-
    aggregate2.max_distance - self.cell_radius
    outer_cells=np.empty((0,3))
    cm=aggregate1.masses_centre
    for ii in xrange(aggregate1.Np):
        cell=aggregate1.cells[ii]
        dist=math.sqrt((cell[0]-
cm[0])**2 + (cell[1] - cm[1])**2 + (cell[2] - cm[2])**2)
        if dist > inner_radius:
            outer_cells=np.vstack((outer_cells,cell))
    return outer_cells

def overlappable_cells_list_CL(self, aggregate_a, aggregate_b,
                                max_distance_b, masses_centre_b):

```

```

d=(max_distance_b+self.cell_radius)**2
aggregate_a_CL=pv.Matrix(np.float32(aggregate_a))
all_cb=np.array(np.shape(aggregate_a))
all_cb[:]=masses_centre_b
dif3= aggregate_a_CL
distances= np.empty((np.shape(aggregate_a)[0],1))
distances=pv.Matrix(np.float32(distances))
cb=pv.Matrix(np.float32(all_cb))
dif3= pv.InplaceSub(dif3,cb)
distances=dif3[:,0]*dif3[:,0]+dif3[:,1]*dif3[:,1]+
    dif3[:,2]*dif3[:,2]
overlappable_cells=np.empty((0,3))
for ii in xrange(len(aggregate_a)):
    if distances[ii].value < d:
        overlappable_cells=np.vstack((overlappable_cells,
            aggregate_a[ii]))
return overlappable_cells

def overlappable_cells_list(self, aggregate_a, aggregate_b,
    max_distance_b, masses_centre_b):

    d=max_distance_b+self.cell_radius

    overlappable_cells=np.empty((0,3))
    cb=masses_centre_b

    for ii in xrange(len(aggregate_a)):
        cell=aggregate_a[ii]

        dist=math.sqrt((cell[0]-
            cb[0])**2 + (cell[1] - cb[1])**2 +
            (cell[2] - cb[2])**2)
        if dist < d:
            overlappable_cells=np.vstack(
                (overlappable_cells,cell))
    return overlappable_cells

def check_overlap_lists(self, lista_a, lista_b):
    #comprueba si hay dos celulas que ocupan mismo volumen
    #entre las exteriores

    dist=0
    overlapped= False
    for a in xrange(len(lista_a)):
        for b in xrange(len(lista_b)):
            dist = math.sqrt((lista_a[a][0]-
                lista_b[b][0])**2 + (lista_a[a][1] -
                lista_b[b][1])**2 + (lista_a[a][2] -
                lista_b[b][2])**2)

            if dist < 2.0 * self.cell_radius:
                overlapped= True
    return overlapped

def get_angle(self, M_1):

```

```

C1_P1=distance.euclidean(M_1, self.aggregate1.masses_centre)
C1_C2= self.gamma_distance
C2_P1=distance.euclidean(M_1, self.aggregate2.masses_centre)
angle=math.acos((C1_P1**2 + C1_C2**2 - C2_P1**2)/(2 * C1_P1 *
C1_C2))
return angle

def get_angle_prime(self, M_2,M_1):

    C2_P2=distance.euclidean(M_2, self.aggregate2.masses_centre)
    C1_C2= self.gamma_distance
    P1_C1=distance.euclidean(M_1, self.aggregate1.masses_centre) #
    #1a misma distancia que C
    #2 P1prime
    angle=math.acos((C2_P2**2 + C1_C2**2 - P1_C1**2)/(2.0 *
C2_P2 * C1_C2))
    return angle

def distance(self, v1,v2):

    dist=distance.euclidean(v1,v2)
    return dist

def get_angle_delta_prime(self,M_2_rotated, M_1):
    delta_prime= math.acos((self.distance(M_2_rotated,
self.aggregate1.masses_centre)**2 + self.distance(
M_1, self.aggregate1.masses_centre)**2 -
(2.0*self.cell_radius)**2) / (2.0 * self.distance
(M_2_rotated,
self.aggregate1.masses_centre)
* self.distance(M_1,
elf.aggregate1.masses_centre)))
    return delta_prime

def get_angle_delta(self,M_2_rotated, M_1):

    delta= math.acos((self.distance(M_2_rotated,
self.aggregate1.masses_centre)**2
+ self.distance(M_1, self.aggregate1.masses_centre)**2 -
(self.distance(M_1_rotated, M_1)**2) / (2.0 *
self.distance(M_2_rotated,
self.aggregate1.masses_centre) * self.distance(M_1,
self.aggregate1.masses_centre)))
    return delta

def angle_2_vectors(self,v1,v2): #calcula angulo entre dos
#vectores dados

    angle= math.acos((np.dot(v1,v2)/(np.linalg.norm(v1) *
np.linalg.norm(v2))))
    return angle

def get_theta(self, M_1, M_2):

```

```

v_1= M_1 - self.aggregate2.masses_centre
theta=angle_2_vectors(v_1, M_2)
return theta

def plane_vector(self,a, b, c):

    return np.cross(b-a, b-c)

def unit_vector(self, v):

    return np.array([v[0]/np.linalg.norm(v), v[1]/
                    np.linalg.norm(v), v[2]/
                    np.linalg.norm(v)])

#####
def q_merge(self):
    #funcion principal de la clase que efectua la union de los
    #dos floculos menores

    not_valid=True
    counter=0
    while not_valid:
        if self.aggregate1.max_distance + self.aggregate2.
            max_distance + 2.0*
            self.cell_radius >= self.gamma_distance:

            not_valid=False
        else:
            counter+=1
            self.aggregate2.add_cells(1)
            #print "counter"+np.str(counter)
    not_merged=True
    list_1=self.create_intersection_list(

self.aggregate1,self.aggregate2)

    while not_merged:
        self.move_floc()
        list_2=self.create_intersection_list(

self.aggregate2,self.aggregate1)
        #print "list_2" + np.str(list_2)
        while True:

            M_1= list_1[np.random.randint(0, np.shape(list_1)
                                         [0])]
            M_2= list_2[np.random.randint(0, np.shape(list_2)[
                                         0])]
            if self.distance(M_1,
                self.aggregate1.masses_centre) +
                self.distance(M_2, self.aggregate2.
                    masses_centre)+ 2.0 * self.cell_radius >=
                self.gamma_distance:

                break

```

```

try:
    rotation_vector_1=np.cross(M_2-
        self.aggregate2.masses_centre,self.
        aggregate1.masses_centre -
        self.aggregate2.masses_centre)
    rotation_vector_1=rotation_vector_1/
        np.linalg.norm(rotation_vector_1)
    angle_M_1 =self.angle_2_vectors(M_2-
self.aggregate2.masses_centre,
        self.aggregate1.masses_centre -
        self.aggregate2.masses_centre)
    angle_prime=self.angle_2_vectors(self.aggregate1.
        masses_centre-self.aggregate2.masses_centre,
        M_2-self.aggregate2.masses_centre)
    rotation_angle= angle_M_1 -angle_prime
    M_2_ext = np.concatenate((M_2,[1]), axis=0)
    rotation_matrix_1=self.rotation_matrix(
        rotation_angle,rotation_vector_1,
        self.aggregate2.masses_centre)
    M_2_rotated=np.dot(rotation_matrix_1, M_2_ext)
    M_2_rotated=M_2_rotated[:3]
    rotated_cells_2=[np.dot(rotation_matrix_1,
        np.concatenate((a,[1]), axis=0)) for a in
        self.aggregate2.cells]
    rotated_cells_2=[a[:3] for a in rotated_cells_2]
    rotated_masses_centre_2=np.dot(rotation_matrix_1,
        np.concatenate((self.aggregate2.masses_centre,
        [1]), axis=0))
    rotated_masses_centre_2=rotated_masses_centre_2[
        :3]
    delta=self.angle_2_vectors(M_2_rotated-
        self.aggregate1.masses_centre,
        M_1-self.aggregate1.masses_centre)
    ang=(distance.euclidean(M_2_rotated,
        self.aggregate1.masses_centre)**2 +
        distance.euclidean(M_1,
        self.aggregate1.masses_centre)**2 -
        (2.0 * self.cell_radius)**2) /
        (2.0*distance.euclidean(M_2_rotated,
        self.aggregate1.masses_centre)*
        distance.euclidean(M_1,
        self.aggregate1.masses_centre))
    #print "ang: %s" %ang
    delta_prime= math.acos(ang)
    rotation_angle_2=delta-delta_prime
    rotation_vector_2= np.cross(M_1-
        self.aggregate1.masses_centre,
self.aggregate2.masses_centre-
        self.aggregate1.masses_centre )
    rotation_matrix_2=self.rotation_matrix(
        rotation_angle_2,rotation_vector_2,
        self.aggregate1.masses_centre)
    M_1_ext = np.concatenate((M_1,[1]), axis=0)
    M_1_rotated=np.dot(rotation_matrix_2, M_1_ext)
    M_1_rotated=M_1_rotated[:3]
    rotated_cells_1=[np.dot(rotation_matrix_2,
        np.concatenate((a,[1]), axis=0))
for a in self.aggregate1.cells]
    rotated_cells_1=[a[:3] for a in rotated_cells_1]

```

```

list_overlap_1 =self.overlappable_cells_list(
    rotated_cells_1,rotated_cells_2,
    self.aggregate2.max_distance,
    self.aggregate2.masses_centre)
list_overlap_2 =self.overlappable_cells_list(
    rotated_cells_2,
    rotated_cells_1, self.aggregate1.max_distance,
    self.aggregate1.masses_centre)
overlapped= self.check_overlap_lists(
    list_overlap_1,list_overlap_2)

if overlapped:
    max_rotations=25
    ii=0
    while ii < max_rotations:
        rotation_try_angle= math.pi*
            random.uniform(-1,1)
        rotation_matrix_test=self.rotation_matrix(
            rotation_try_angle, M_2_rotated -
            self.aggregate2.masses_centre,
            M_2_rotated)
        rotated_cells_2_temp=[np.dot(
            rotation_matrix_test,
            np.concatenate(
(a,[1]), axis=0))

                for a in rotated_cells_2]
        rotated_cells_2_temp=[a[:3] for a in
            rotated_cells_2]
        list_overlap_1 =self.
            overlappable_cells_list(
            rotated_cells_1,
            rotated_cells_2_temp,
            self.aggregate2.max_distance,
            self.aggregate2.masses_centre)
        list_overlap_2 =self.
            overlappable_cells_list(
            rotated_cells_2_temp,
            rotated_cells_1,
            self.aggregate1.max_distance,
            self.aggregate1.masses_centre)
        overlapped= self.check_overlap_lists(
            list_overlap_1,
            list_overlap_2)
        if overlapped == True:
            ii+=1
        else:
            composite_aggregate=Aggregate(self.kf,
                self.df, len(rotated_cells_1)+
                len(rotated_cells_2_temp),
                cells=np.concatenate(
                    (rotated_cells_1,
rotated_cells_2_temp), axis=0 ))
                not_merged = False
                return composite_aggregate
    else:
        composite_aggregate=Aggregate(self.kf, self.df
            , len(rotated_cells_1)+

```



```

        len(rotated_cells_2),
        cells=np.concatenate((rotated_cells_1,
                               rotated_cells_2), axis=0 ))
    not_merged = False

    return composite_aggregate
    #print "intentando acople.."
except ValueError:
    pass

```

CROP TO LENGTH_2

La clase *crop_TO_LENGTH_2* contiene un algoritmo de crecimiento de agregados diseñado e implementado para construir un flóculo aleatoriamente a partir de valores consignados de la dimensión fractal y el diámetro de Feret máximo. El algoritmo lleva a cabo la combinación iterativa de flóculos de menor tamaño. Como la construcción de los flóculos es aleatoria, no se puede conocer a priori el tamaño del agregado resultante, con lo que el proceso de crecimiento ha de ser iterativo, puesto que, en ocasiones, el flóculo final resultante presenta un diámetro de Feret máximo superior a la suma del diámetro de consigna y el margen de tolerancia fijado. El algoritmo se diseñó al objeto de ahorrar tiempo de computación mediante el *reciclado* de uno de los flóculos que ha dado lugar a un agregado de tamaño no válido superior al consignado.

```

# -*- coding: utf-8 -*-
"""
Created on Tue Oct 04 10:49:12 2016
@author: patricio
"""

from __future__ import division
#import linecache
import sys
import random, math
from random import jumpahead
from random import uniform, random
import numpy as np
from numpy import cross, eye, dot
import math
import numpy
import copy
from scipy.spatial import distance
from scipy.linalg import expm3, norm
from scipy import stats
from aggregate6 import Aggregate
from merger_2_no_parallel import Floccs_merger
import os
import pickle
from scipy import weave

class crop_TO_LENGTH_2:

```

```

def __init__(self, k,d,length, base_floc_Np, tolerance=None):

    self.cell_radius=np.float64(x=40.0)
    self.scale= 1.5/self.cell_radius
    self.kf=k
    self.df=d
    self.base_floc_Np = base_floc_Np
    self.target_length=length
    self.lower_value=0
    self.upper_value =0
    self.current_floc_1=Aggregate(self.kf, self.df,
                                  self.base_floc_Np)
    self.current_floc_1.add_cells()
    self.current_floc_2=Aggregate(self.kf, self.df,
                                  self.base_floc_Np)
    self.current_floc_2.add_cells()
    self.tries=0
    if tolerance!= None:
        self.tolerance=tolerance
    else:
        self.tolerance=0 #no lo abordo

def reset(self): #vuelve el proceso al comienzo si la combinacion
#de floculos no es exitosa porque se supera la tolerancia o no se
#encuentra combinación geométrica compatible

    self.current_floc_1=Aggregate(self.kf, self.df,
                                  self.base_floc_Np)
    self.current_floc_1.add_cells()
    self.current_length_1= self.current_floc_1.length
    self.current_floc_2=Aggregate(self.kf, self.df,
                                  self.base_floc_Np)
    self.current_floc_2.add_cells()
    self.tries = 0

def crop(self):

    if self.tolerance != 0:
        while True: #tries <10:
            try:
                merger=Flocs_merger(self.current_floc_1,
                                    self.current_floc_2)
                composite_aggregate_1 = merger.q_merge()
                #print "composite 1"
                self.current_floc_1 = composite_aggregate_1
                #print "current_floc_1"
                floc_length=self.current_floc_1.length
                #print "current floc length: %d" % (floc_length)
                if floc_length >= self.target_length -
                    self.tolerance and floc_length <
                    self.target_length +self.tolerance:
                    return self.current_floc_1
                    break
                elif floc_length < self.target_length -
                    self.tolerance :

                    tries = 0

```

```

        while tries < 5:
            composite_aggregate_2 = merger.q_merge()
            tries+=1 #incrementa el Contador de
#intentos de coalescencia
            if composite_aggregate_2.length +
                self.current_floc_1.length <self.
                    target_length -
                        self.tolerance:
                            self.current_floc_2 =
                                composite_aggregate_2
                                    break
            else:
                self.reset()
                self.tries +=1
        except ValueError:
            pass

    if self.tolerance == 0:
        tries = 0
        while True:#tries <10:

            merger=Flocs_merger(self.current_floc_1,
                                self.current_floc_2)
            composite_aggregate_1 = merger.q_merge()
            #print "composite 1"
            self.current_floc_1 = composite_aggregate_1
            #print "current_floc_1"
            floc_length=self.current_floc_1.length
            #print "floc length: %d" % (floc_length)
            if floc_length >= self.lower_value and floc_length <
                self.upper_value :

                return self.current_floc_1
                break

            elif floc_length < self.lower_value :
                composite_aggregate_2 = merger.q_merge()
                self.current_floc_2 = composite_aggregate_2
            else:
                self.reset()
                tries +=1

    def measure_length(self, floc):

        outer_cells=floc.outer_cells
        furthest=floc.furthest_cell
        distancias=np.empty((np.shape(outer_cells)[0],1))
        for ii in xrange(len(outer_cells)):
            distancias[ii]= distance.euclidean(furthest, outer_cells[ii]
)
        longitud= np.max(distancias)
        longitud=(longitud+2*self.cell_radius)*self.scale#3.5/10.0
        indice = np.argmax(distancias)
        return longitud, indice,distancias

```

FLOC PROJECTOR

La clase *floc_projector* toma un flóculo, lleva a cabo una sedimentación virtual en base a una estimación del área de proyección máxima del mismo, obtiene su proyección bidimensional en el plano y devuelve esta última en la forma de una matriz de bits binaria. Estas proyecciones se procesan posteriormente mediante ImageJ para determinar la dimensión fractal en 2D de cada flóculo.

```
# -*- coding: utf-8 -*-
"""
Created on Thu Oct 27 14:56:07 2016
@author: patricio
"""

from __future__ import division
import linecache
import sys
import random, math
from random import jumpahead

import numpy as np
import math
from scipy.spatial import distance
from scipy import stats
from matplotlib import pyplot as plt
import scipy
import matplotlib.image as mpimg
import cv2 as cv
import pickle
from aggregate6 import Aggregate
import patri_math as p_math
import sys

class flocc_projector:

    def __init__(self, flocc):

        self.flocc= flocc
        self.cell_radius=flocc.cell_radius
        self.length_canvas_x=0#np.int(flocc.max_distance * 3)
        self.length_canvas_y=0#np.int(flocc.max_distance * 3)
        self.projection = None
        self.max_area = None
        self.prueba = None
        self.angles = (2*np.pi)/np.arange(1,360,1)
        self.settled_flocc=None
        self.colour=int(x=1)

    def get_flocc_projection(self):

        self.settle_flocc()
        self.get_max_area_()

    def settle_flocc(self):
```

```

self.settled_floc=Aggregate(self.floc.kf, self.floc.df,
                             self.floc.Np,
                             self.rotate_major_axis(self.floc))

def get_projection(self,cluster,masses_centre, furthest, outer):

    projected_cluster=np.empty((np.shape(cluster)[0],2))
    p_outer=np.empty((np.shape(outer)[0],2))
    for ii in xrange(np.shape(cluster)[0]):
        projected_cluster[ii]=(tuple((np rint(cluster[ii][0]),
                                       np rint(cluster[ii][1]))))
    for ii in xrange(np.shape(p_outer)[0]):
        p_outer[ii]=(tuple((np rint(outer[ii][0]),
                              np rint(outer[ii][1]))))
    p_masses_centre=tuple((np rint(masses_centre[0]),
                           np rint(masses_centre[1]))))
    p_furthest=tuple((np rint(furthest[0]),np rint(furthest[1]))))

    return projected_cluster,p_masses_centre, p_furthest, p_outer

def get_horizontal_pos(self,projected_cluster): #funcion para
#obtener los limites
#horizontales del agregado

###encuentra x min, xmax, y min, y max
x_max = 0
x_min = 0#np.shape(projected_cluster)[1]
for column in xrange (np.shape(projected_cluster)[1]-1,
                      0,-1):
    for row in xrange (np.shape(projected_cluster)[0]):
        if projected_cluster[row][column] != 0:
            x_max = column
            break
for column in xrange (np.shape(projected_cluster)[1]):
    for row in xrange (np.shape(projected_cluster)[0]):
        if projected_cluster[row][column] != 0 :
            x_min = column
            break
print "xmin: "+np.str(x_min)+ " xmax: "+np.str(x_max)
return x_min, x_max

def get_vertical_pos(self,projected_cluster):      #limites
#verticales

y_max=0
y_min=np.shape(projected_cluster)[1]
#y_max
for row in xrange (np.shape(projected_cluster)[0]-1,0,-1):
    for column in xrange (np.shape(projected_cluster)[1]):
        if projected_cluster[row][column] != 0 :
            y_max = row
            break
for row in xrange (np.shape(projected_cluster)[0]):
    for column in xrange (np.shape(projected_cluster)[1]):

```

```

        if projected_cluster[row][column] != 0 :
            y_min = row
            break
    print "y: "+np.str(y_min)+ " ymax: "+np.str(y_max)
    return y_min, y_max

def center_floc(self,projected_cluster, p_masses_centre, p_furthest,
p_outer):
    #mueve la proyeccion del floculo al centro del tapiz de proyeccion

#
    xmin,xmax=self.get_horizontal_pos(projected_cluster)
    xmin = min([x[0] for x in projected_cluster]) -
        self.cell_radius
    xmax = max([x[0] for x in projected_cluster]) +
        self.cell_radius
    ymin = min([x[1] for x in projected_cluster]) -
        self.cell_radius
    ymax = max([x[1] for x in projected_cluster]) +
        self.cell_radius
    xlength=xmax-xmin
    self.length_canvas_x = int(x=xlength*1.5)
    xcenter=xmin+xlength//2
    ylength=ymax-ymin
    self.length_canvas_y=int(x=np.ceil(ylength*1.5))
    ycenter=ymin+ylength//2
    xtrans= self.length_canvas_x//2-xcenter
    ytrans= self.length_canvas_y//2-ycenter
    centered_projected=[]
    c_outer=[]
    for ij in xrange(len(projected_cluster)):
        centered_projected.append(tuple(map(
            lambda x, y: x + y, projected_cluster[ij],
            (xtrans,ytrans))))

    for ij in xrange(len(p_outer)):
        c_outer.append(tuple(map(lambda x, y: x + y,
            p_outer[ij], (xtrans,ytrans))))
    centered_projected=np.asarray(centered_projected)
    centered_projected.astype(int)
    xmin = min([x[0] for x in centered_projected]) -
        self.cell_radius
    xmax = max([x[0] for x in centered_projected]) +
        self.cell_radius
    ymin = min([x[1] for x in centered_projected]) -
        self.cell_radius
    ymax = max([x[1] for x in centered_projected]) +
        self.cell_radius
    c_masses_centre= np rint(p_masses_centre[0]+xtrans),
        np rint(p_masses_centre[1]+ ytrans)
    c_furthest=np rint(p_furthest[0]+xtrans), np rint(
        p_furthest[1]+ytrans)
#
    print c_furthest
#
    print centered_projected
    return centered_projected, c_masses_centre, c_furthest, c_outer

def generate_floc_matrix(self, centered_projected_floc,

```

```

        masses_centre, furthest, c_outer):
    img= np.zeros((self.length_canvas_y, self.length_canvas_x),
        np.uint8) # crea la matriz
#de proyeccion en blanco con las medidas del tapiz
    for ii in xrange(len(centered_projected_floc)):
        cv.circle(img,(np.int(x=centered_projected_floc[ii][0]),
            np.int(x=centered_projected_floc[ii][1])),
            int(self.cell_radius), self.colour,
            thickness=-1, lineType=4, shift=0)
"""
#proyecta las celulas como discos
#comprobacion de la proyeccion del centro de masas, la celula mas lejana
y celulas exteriores del floculo

#cv.circle(img,(np.int(x=masses_centre[0]),np.int(x=masses_centre[1]))
    , #int(self.cell_radius), int(x=11), thickness=-
    1, lineType=4, shift=0) #
#cv.circle(img,(np.int(x=furthest[0]),np.int(x=furthest[1])),
    int(self.cell_radius), int(x=7), thickness=-
    1, lineType=4, shift=0) #
#for ii in xrange(len(c_outer)):
    cv.circle(img,(np.int(x=c_outer[ii][0]),
#np.int(x=c_outer[ii][1])), int(self.cell_radius), int(x=10),
    thickness=-1, lineType=4, shift=0) #
"""

    return img

def rotate_major_axis(self,floc):

    rotated_cells = np.empty((floc.Np, 3),dtype=np.float64)
    moved_cells = np.empty((floc.Np, 3),dtype=np.float64)
    vector_displacement = np.array( [floc.masses_centre[0],
        floc.masses_centre[1],0] )
        - np.array( floc.masses_centre )
    furthest_semi_projected = vector_displacement +
        floc.furthest_cell
    vector_1 = furthest_semi_projected -
        np.array( [floc.masses_centre[0],
            floc.masses_centre[1],0] )
    vector_2 = np.array( [floc.furthest_cell[0],
        floc.furthest_cell[1],0]) -
        np.array( [floc.masses_centre[0],
            floc.masses_centre[1],0])
    projection_angle = p_math.angle_2_vectors(vector_1, vector_2)
    rotation_vector = np.cross(vector_1, vector_2)
    rotation_vector = rotation_vector/np.linalg.norm(
        rotation_vector)
    rot_matrix = p_math.rotation_matrix(projection_angle,
        rotation_vector, np.array( [floc.masses_centre[0],
            floc.masses_centre[1],0] ))
    moved_cells=np.asarray([a + vector_displacement for a in
        floc.cells])
    temp_cell=np.empty((1,4))
    for ii in xrange(floc.Np):
        temp_cell = (np.dot(rot_matrix ,np.concatenate(
            (moved_cells[ii],[1]), axis=0)))
        rotated_cells[ii,:]=temp_cell[:3]

```

```

        rotated_cells=np.asarray([a - vector_displacement for a in
                                rotated_cells])
    return rotated_cells

def get_max_area(self): #funcion de estimacion del area de la
proyeccion en pixels

    major_axis = self.settled_floc.furthest_cell -
self.settled_floc.masses_centre
    rotated_floc_temp = np.empty((len(self.settled_floc.cells),4))

    rotated_floc= np.empty((len(self.settled_floc.cells),3))
    rotated_outer=np.empty((len(self.settled_floc.outer_cells),3))

    rotated_outer_temp = np.empty((len(self.settled_floc.outer_cells
),4))
    projected, p_mass_centre, p_furthest, p_outer = self.get_project
ion(self.settled_floc.
cells.astype(int), self.settled_floc.masses_centre,

self.settled_floc.furthest_cell, self.settled_floc.outer_cells)

    centered_projected, c_masses_centre, c_furthest, c_outer = self.
center_floc(projected,
p_mass_centre, p_furthest, p_outer)
    floc_matrix=self.generate_floc_matrix(centered_projected,c_masse
s_centre, c_furthest,
c_outer )
    max_area= np.bincount(floc_matrix.ravel())[self.colour]
    max_matrix=floc_matrix
    angle_max=0
    print "area a cero: "+np.str(max_area)
    for angle in xrange(1,len(self.angles),1):
        rot_matrix = p_math.rotation_matrix(self.angles[angle],major
_axis,

self.settled_floc.masses_centre)

        for cell in xrange(len(self.settled_floc.cells)):
            rotated_floc_temp[cell] = np.dot(rot_matrix,np.concatena
te((self.settled_floc.
cells[cell],[1]), axis=0))
            rotated_floc = np.around(rotated_floc_temp[:, :3],decimals=0)
            .astype(int)
            for cell in xrange(len(self.settled_floc.outer_cells)):
                rotated_outer_temp[cell] = np.dot(rot_matrix,np.concaten
ate((self.settled_floc
.outer_cells[cell],[1]), axis=0))

            rotated_outer = np.around(rotated_outer_temp[:, :3],decimals=
0).astype(int)
            rotated_masses_centre= np.dot(rot_matrix,np.concatenate(
(self.settled_floc.masses_centre,[1]), axis=0))
            rotated_masses_centre= np.around(rotated_masses_centre[:3],d
ecimals=0).astype(int)
            rotated_furthest= np.dot(rot_matrix,np.concatenate(
(self.settled_floc.furthest_cell,[1]), axis=0))
            rotated_furthest= np.around(rotated_furthest[:3],decimals=0)
            .astype(int)

```



```

        projected, p_mass_centre, p_furthest, p_outer =self.get_projection(rotated_floc,

        rotated_masses_centre,rotated_furthest, rotated_outer)
        centered_projected, c_mass_centre, c_furthest, c_outer =

        self.center_floc(projected, p_mass_centre, p_furthest, p_outer)

        floc_matrix=self.generate_floc_matrix(centered_projected,c_masses_centre,
        c_furthest, c_outer)
        area_temp = np.bincount(floc_matrix.ravel())[self.colour]

        if area_temp > max_area:
            max_area = area_temp
            max_matrix = floc_matrix
            angle_max=1/self.angles[angle]/(2*np.pi)
        print "area max: "+np.str(max_area)
        print "length_x: "+np.str(self.length_canvas_x)
        print "length_y: "+np.str(self.length_canvas_y)
        print "angle: "+np.str(angle_max)
        self.max_area = max_area
        self.projection = max_matrix

    def measure_length(self, floc):

        outer_cells=floc.outer_cells
        furthest=floc.furthest_cell
        distancias=np.empty((np.shape(outer_cells)[0],1))
        for ii in xrange(len(outer_cells)):
            distancias[ii]= distance.euclidean(furthest, outer_cells
[ii])
        #         print distancias
        longitud= np.max(distancias)
        longitud=longitud+2*self.cell_radius#*3.5/10.0
        indice = np.argmax(distancias)
        #         print "indice original: "+np.str(indice)
        return longitud, indice,distancias
        #         distance.euclidean([446.8/712.0, 377.0/522.0], [345.0/712.0
, 116.0/522.0])

```

ROTATOR

La clase *rotator* tiene como función generar un número dado de rotaciones aleatorias de un flóculo en el espacio y obtener la proyección correspondiente a cada posición obtenida sobre un plano. Esta proyección se transforma en una matriz como mapa de bits (*raster*). Posteriormente, de las proyecciones se obtienen los datos de distribución de tamaños virtuales mediante un programa que simula el proceso de escaneo de la sonda láser por reflexión de haz enfocado FBRM.

```

# -*- coding: utf-8 -*-
"""
Created on Fri May 06 11:53:57 2016

```

```
@author: plopez
"""

from __future__ import division
import linecache
import sys
import random
import math

from random import jumpahead
from time import time
import numpy as np
import math
from scipy.spatial import distance
from scipy import signal
from scipy import stats
from sklearn.linear_model import LinearRegression
from matplotlib import pyplot as plt
import scipy
import matplotlib.image as mpimg
import cv2 as cv
import pickle
import os
import platform
import path_loader as ploader #carga las rutas según sea la estación de
trabajo o EOL0

class rotator:

    def __init__(self, **kwargs):

        file_handler= open(os.path.join(ploader.intervals_path, 'intervals.p'), 'rb')
        self.intervals=pickle.load( file_handler)
        file_handler.close()
        file_handler= open(os.path.join(ploader.intervals_path, 'mid_points.p'), 'rb')
        self.mid_points=pickle.load(file_handler)
        file_handler.close()
        self.cell_radius=np.int(40)
        self.shading = 0.
        self.beam_width = 10
        self.scale= 1.5/self.cell_radius
        self.real_laser_width = 10
        self.beam_width = np.int(np.floor(self.real_laser_width/self.scale)) #el haz de las
        self.length_canvas_x=0#np.int(floc.max_distance * 3)
        self.length_canvas_y=0#np.int(floc.max_distance * 3)
        self.x_span=np.int(kwargs.pop('x_span')) #np.floor(x_span/self.scale)
        self.rotations=kwargs.pop('rotations')
        self.spectra=np.empty((self.rotations,200), dtype=np.float32)
        self.chord_lengths=[]
        floc=kwargs.pop('floc')
        self.cells=floc.cells
        self.masses_centre=floc.masses_centre
```

```

self.prueba=[]
self.projections=[]
self.kernel = np.ones((6,6),np.float32)/36
self.colour=np.int(x=3)
self.distribution=np.zeros(200)
self.mean_dist=np.zeros(200)
self.smoothed=np.empty((self.rotations,200), dtype=np.float32)
self.rayas=[]
self.externally_created = False

def suavizar(self):

    for xx in xrange(self.rotations):
        self.smoothed[xx,:200]=scipy.signal.medfilt(self.spectra[xx]
[:],13)

def gen_angle(self):

    jumpahead(10)
    x=random.uniform(-1,1)
    y=random.uniform(-1,1)
    z=random.uniform(-1,1)
    magnitud=math.sqrt(x**2+y**2+z**2)
    return np.array([x/magnitud,y/magnitud,z/magnitud])

def rotation_matrix(self,angle, direction, point=None):

    sina = math.sin(angle)
    cosa = math.cos(angle)
    direction = self.unit_vector(direction[:3])
    R = np.diag([cosa, cosa, cosa])
    R += np.outer(direction, direction) * (1.0 - cosa)
    direction *= sina
    R += np.array([[ 0.0,          -direction[2],  direction[1]],
[ direction[2], 0.0,          -
direction[0]],
[ -direction[1], direction[0],  0.0]])
    M = np.identity(4)
    M[:3, :3] = R
    if point is not None:
        point = np.array(point[:3], dtype=np.float64, copy=False)
        M[:3, 3] = point - np.dot(R, point)
    return M

def unit_vector(self, v):

    return np.array([v[0]/np.linalg.norm(v), v[1]/np.linalg.norm(v),
v[2]/np.linalg.norm(v)])

def get_projection(self,cluster):

    projected_cluster=np.empty((np.shape(cluster)[0],2))
    for ii in xrange(np.shape(cluster)[0]):
        projected_cluster[ii]=(tuple((np rint(cluster[ii][1]),

```

```

np rint(cluster[ii][0]))))
    return projected_cluster

def get_horizontal_pos(self,projected_cluster):

    x_max = 0
    x_min = np.shape(projected_cluster)[1]

    for column in xrange (np.shape(projected_cluster)[1]-1, 0,-1):
        for row in xrange (np.shape(projected_cluster)[0]):
            if projected_cluster[row][column] != 0:
                x_max = column
                break

    for column in xrange (np.shape(projected_cluster)[1]):
        for row in xrange (np.shape(projected_cluster)[0]):
            if projected_cluster[row][column] != 0 :
                x_min = column
                break
    print "xmin: "+np.str(x_min)+ " xmax: "+np.str(x_max)
    return x_min, x_max

def get_vertical_pos(self,projected_cluster):

    y_max=0
    y_min=np.shape(projected_cluster)[0]
    for row in xrange (np.shape(projected_cluster)[0]-1,0,-1):
        for column in xrange (np.shape(projected_cluster)[1]):
            if projected_cluster[row][column] != 0 :
                y_max = row
                break

    for row in xrange (np.shape(projected_cluster)[0]):
        for column in xrange (np.shape(projected_cluster)[1]):
            if projected_cluster[row][column] != 0 :
                y_min = row
                break
    print "y: "+np.str(y_min)+ " ymax: "+np.str(y_max)
    return y_min, y_max

def center_floc(self,projected_cluster):

    xmin = min([x[1] for x in projected_cluster]) - self.cell_radius
    xmax = max([x[1] for x in projected_cluster]) + self.cell_radius
    ymin = min([x[0] for x in projected_cluster]) - self.cell_radius
    ymax = max([x[0] for x in projected_cluster]) + self.cell_radius

    xlength=xmax-xmin
    ylength=ymax-ymin

    if xmin<0:
        xtrans=np.abs(xmin)+2*self.cell_radius
    else:

```

```

        xtrans=0
    if ymin<0:
        ytrans=np.abs(ymin)+2*self.cell_radius
    else:
        ytrans=0
    self.length_canvas_x = int(xlength+11*self.cell_radius)
    self.length_canvas_y=int(ylength+11*self.cell_radius)
    centered_projected=[]
    for ij in xrange(len(projected_cluster)):
        centered_projected.append(tuple(map(lambda x,y: x + y, proje
cted_cluster[ij],
        (ytrans,xtrans))))
    centered_projected=np.asarray(centered_projected)
    centered_projected.astype(int)
    return centered_projected

def generate_floc_matrix(self, centered_projected_floc):

    img= np.zeros((self.length_canvas_y,self.length_canvas_x), np.ui
nt8)
    for ii in xrange(len(centered_projected_floc)):
        cv.circle(img,(np.int(x=centered_projected_floc[ii][1]),
        np.int(x=centered_projected_floc[ii][0])),
        int(self.cell_radius), self.colour, thickness=-1, lineType=4, shift=0)
    return img

def get_matrix_limits(self,floc_matrix):

    minx=np.size(floc_matrix,1)-1
    maxx=np.size(floc_matrix,1)-1
    for xx in xrange (np.size(floc_matrix,1)):
        if any(floc_matrix[:,xx]==self.colour):
            minx=xx
            break
    for xx in xrange (np.size(floc_matrix,1)-1,0,-
1):
        if any(floc_matrix[:,xx]==self.colour) :
            maxx=xx
            break
    #print "min, max: "+np.str(minx)+" "+np.str(maxx)
    return minx, maxx

def rotate_floc(self,floc):

    rotation_direction=self.gen_angle()
    phi=math.pi*random.uniform(-1,1)
    rot_matrix = self.rotation_matrix(phi, rotation_direction,self.m
asses_centre )
    rotated_cells=[]
    temp_cell=np.empty((1,4))
    for ii in xrange(len(floc)):
        temp_cell = (np.dot(rot_matrix ,np.concatenate((floc[ii],[1]
), axis=0)))
        rotated_cells.append(temp_cell)
    return rotated_cells

```

```
def generate_projections(self):# metodo principal de la clase

    y_margin= self.beam_width//2
    for rot in xrange(self.rotations):

        rotated_floc= self.rotate_floc(self.cells)
        projected=self.get_projection(rotated_floc)
        centered_projected=self.center_floc(projected)
        floc_matrix=self.generate_floc_matrix(centered_projected)

        self.projections.append(floc_matrix)
```

SAVE PROJECTIONS

El programa que se incluye a continuación se creó para generar una población de proyecciones aleatorias para cada agregado. El programa toma un agregado, lo rota aleatoriamente un número de veces consignado (cien), y almacena cada proyección en un archivo comprimido del tipo HDF5 (*Hierarchical Data Format*). En el código, los valores de rpm se refiere a los grupos de tamaño de flóculos medidos sobre las imágenes de microscopía confocal.

```
# -*- coding: utf-8 -*-
"""
Created on Sat Mar 11 21:30:26 2017

@author: patricio
"""

import os
import pickle
from collect_table_E0 import collect_table
import aggregate6
import path_loader as ploader
from floc_rotator_E0 import rotator
from joint_distribution_2 import joint_distribution_2
import h5py
rpms=[500,350,200]
fds=[1.3, 1.4, 1.5, 1.6,1.7,1.8,1.9,2.0,2.1]#[1.3]#
rotations=100
path_rot=ploader.samples_path+'/rotaciones'

def gen_rot(fl):

    rot=rotator(floc= fl, rotations= 100, x_span=30)
    rot.generate_projections()
    return rot.projections

for r in rpms:
    rpm=str(r)

    for f in fds:
        fd=str(f)

        cl=collect_table(rpm,f)
        flocs=cl.flocs
```

```

for f1 in xrange(len(flocs)):
    projections=gen_rot(flocs[f1])

    new_dir=os.path.join(path_rot,'rot_'+rpm+'_'+fd)
    if not os.path.isdir(new_dir):
        os.mkdir(new_dir)

    print "floc %d" % f1

    for r in xrange(rotations):
        num_floc='{03d}'.format(f1)
        num_rot='{03d}'.format(r)
        name= 'rotation_fd_'+np.str(fd)+'_'+np.str(rpm)+'floc_'+
num_floc+'_rot_num_'+'+num_rot+'_.hdf5'

        f = h5py.File(os.path.join(new_dir, name),'w')
        grp=f.create_dataset('rotation', data=projections[r], co
mpression='gzip')
        f.close()

```

VIRTUAL SCAN

La clase *virtual_scan* tiene como cometido la realización de un escaneo virtual de las proyecciones de los flóculos de forma que se emule la que realizaría la sonda FBRM y la clasificación de los tamaños de cuerda obtenidos como lo haría el software de dicha sonda. El programa toma una proyección de flóculo, la centra y define aleatoriamente los puntos sobre los que se realiza el escaneo. Este consiste en medir los segmentos de la matriz de imagen en los que se detecta proyección del flóculo. Los segmentos medidos se organizan posteriormente en intervalos de tamaño como los que se obtienen con el software de la sonda FBRM para construir los histogramas que representan las distribuciones de tamaños de cuerda.

```

# -*- coding: utf-8 -*-
"""
Created on Fri May 06 11:53:57 2016
@author: plopez
"""

from __future__ import division
import linecache
import sys
import random
import math

from random import jumpahead
from time import time
import numpy as np
import math
from scipy.spatial import distance
from scipy import signal
from scipy import stats
from sklearn.linear_model import LinearRegression
from matplotlib import pyplot as plt

```

```

import scipy
import matplotlib.image as mpimg
import cv2 as cv
import pickle
import os
import platform
import path_loader as ploader #carga las rutas según sea la estación de
trabajo o EOL0

class v_scan:

    def __init__(self, projection, x_span=30):

        file_handler= open(os.path.join(ploader.intervals_path, 'intervals.p'), 'rb')
        self.intervals=pickle.load( file_handler)
        file_handler.close()
        file_handler= open(os.path.join(ploader.intervals_path, 'mid_points.p'), 'rb')
        self.mid_points=pickle.load(file_handler)
        file_handler.close()
        self.cell_radius=np.int(40)
        self.shading = 0.
        self.beam_width = 10
        self.scale= 1.5/self.cell_radius
        self.real_laser_width = 10 #micras
        self.beam_width = np.int(np.floor(self.real_laser_width/self.scale)) #el haz de las
        self.length_canvas_x=0#np.int(floc.max_distance * 3)
        self.length_canvas_y=0#np.int(floc.max_distance * 3)
        self.x_span=x_span #np.floor(x_span/self.scale)
        self.spectra=np.empty((len(projection),200), dtype=np.float32)
        self.spectrum=np.empty((200,), dtype=np.float32)
        self.chord_lengths=[]
        self.prueba=[]
        self.projection=projection
        self.kernel = np.ones((6,6),np.float32)/36
        self.colour=np.int(x=3)
        self.distribution=np.zeros(200)
        self.mean_dist=np.zeros(200)
        self.smoothed=np.empty((len(projection),200), dtype=np.float32)

        self.rayas=[]
        self.externally_created = False

    def get_matrix_limits(self,floc_matrix):

        minx=np.size(floc_matrix,1)-1
        maxx=np.size(floc_matrix,1)-1
        for xx in xrange (np.size(floc_matrix,1)):
            if any(floc_matrix[:,xx]==self.colour):
                minx=xx
                break
        for xx in xrange (np.size(floc_matrix,1)-1,0,-1):
            if any(floc_matrix[:,xx]==self.colour) :
                maxx=xx

```



```

        break
    return minx, maxx

    def get_scan_positions(self, floc_matrix): #selecciona aleatoriamente
    los puntos de                               #escaneo
    sobre la proyeccion del floculo

        positions=[]
        xmin,xmax=self.get_matrix_limits(floc_matrix)
        length=xmax-xmin
        num_scans=int(x=(length//self.x_span))
        start_position=int(x=(xmin + self.beam_width +np.ceil(length * r
    andom.random()))))
        #elige un punto aleatorio de la anchura del floc a medir
        positions.append(start_position)
        pos=start_position
        for ii in range(1,num_scans):
            pos+=self.x_span
            if pos > xmax:
                difference=pos-xmax
                pos = xmin+difference+self.beam_width
            positions.append(pos)
        return p    ositions

    def line_scan(self):# función principal de la clase

        chord_lengths=[]
        chord_lengths.append(0)
        floc_matrix=self.projection
        pixel_count = 0
        current_chord_length_index=len(self.chord_lengths)-1
        positions= self.get_scan_positions(floc_matrix)
        for position in positions: # escanea todas las posiciones
            repeated_zero=False
            num_zeros=0
            pixel_count = 0
            for yy in xrange(np.size(floc_matrix,0)):
                try:
                    beam_zeros=np.bincount((floc_matrix[yy, position])).
    ravel())[0]
                    #cuenta los pixels del haz virtual en los que
    no se detecta                    #floculo

                except Exception:
                    beam_zeros = 0
                    pass
                if floc_matrix[yy,position-1:position].all() !=0:
                    pixel_count = pixel_count+1 #si se detecta floculo
    se incrementa el                    #contador

                    current_chord_length_index=len(chord_lengths)-1
                    chord_lengths[current_chord_length_index]=chord_len
    gths[
                                current_chord_length_index]+self.scale
                    pixel_count=0
                    repeated_zero=False
                else:
                    pixel_count = 0

```

```

        if not repeated_zero:
            chord_lengths.append(0)
            repeated_zero=True #marca el fin de un
segmento de floculo
        #detectado (cuerda)
        self.spectrum=self.assign_intervals(chord_lengths)
        return chord_lengths

    def assign_intervals(self,chord_lengths_all): #asigna los tamaños
de cuerda obtenidos a cada
intervalo de tamaño

        distribution =np.empty((200,), dtype=np.float32)
        chord_lengths =chord_lengths_all
        for ii in xrange(len(chord_lengths)):
            for jj in xrange(1,len(self.intervals),1):
                if chord_lengths[ii] >= self.intervals[jj-
1] and chord_lengths[ii] < self.intervals[jj]:
                    distribution[jj-1] +=1 #+ self.distribution[jj-1]
        #print "distribution "+np.str(self.distribution[1])
        return distribution

```

COLLECT TABLE

La clase *collect_table_E0* constituye un scrip auxiliar que se emplea para cargar en el sistema los archivos de la clase *aggregate* previamente creados para procesarlos. Los archivos de flóculos se localizan por su grupo de tamaños (rpm) y su dimensión fractal.

```

import os
import numpy as np
import path_loader as ploader

class collect_table_E0:
    def __init__(self,rpm,fd):
        r=np.str(rpm)
        self.fd=np.str(fd)
        path=ploder.flocs_path
        direct= path+ '/flocs_1.5_MICRAS_'+str(r)+'_rpm_df_'+str(fd)+'_.n
py'
        self.flocs = np.load(direct)

```

JOINT DISTRIBUTION

La clase *joint_distribution_2* tiene como cometido la combinación de histogramas de tamaño de un mismo flóculo y la aplicación de un suavizado a dicha distribución.

```

# -*- coding: utf-8 -*-
"""
Created on Mon Feb 13 09:31:02 2017
@author: patricio
"""

```

```

import random, math
import numpy as np
from matplotlib import pyplot as plt
import scipy
from scipy import signal
import matplotlib.image as mpimg
import cv2 as cv
import patri_math as p_math
import pickle
import os
import path_loader as ploader

class joint_distribution_2: #joint_distribution sin normalizar!!!

    def __init__(self, fbrm_distribution=None, filter_size=None):

        if filter_size is not None:
            self.fs=filter_size
        else:
            self.fs=17

        if fbrm_distribution:
            self.reference_distribution= fbrm_distribution
            self.reference_counts=sum (fbrm_distribution)
            self.merged=1
        else:
            self.reference_counts=100
            self.merged=0

        self.joint_dist=np.zeros((200,), dtype=np.float64)
        self.normalized_joint_dist=np.zeros((200,), dtype=np.float64)
        file_handler= open(os.path.join(ploader.intervals_path,"mid_poin
ts.p"), 'rb')
        self.mid_points=pickle.load(file_handler)
        file_handler.close()
        self.smoothed_dist = np.zeros((200,), dtype=np.float64)

    def merge_distribution(self, other_distribution):
        self.merged+=1
        for interval in xrange(200):
            self.joint_dist[interval] = self.joint_dist[interval] +
                other_distribution[interval]/np.float32(self.merged)

    def get_smoothed_distribution(self):

        self.smoothed_dist= scipy.signal.medfilt(self.joint_dist,self.fs
)

    def plot_dist(self):

        fig = plt.figure()
        ax2 = fig.add_subplot(111)
        ax2.title("Distribución de tamaños escaneo virtual")
        ax2.set_autoscale_on(True)
        ax2.set_ylim([0,self.reference_counts])
        ax2.set_xscale('log')
        ax2.set_xlim([1,500])

```

```
ax2.plot(self.mid_points,dist,'-', color='blue')
```

GET SPECTRUM

La función *get_spectrum2* carga las proyecciones en el plano de un flóculo en todos los estados de rotación previamente generados y las somete a un barrido virtual combinando los espectros de tamaño y almacenando el espectro resultante en un archivo de texto .csv (*comma separated values*).

```
#!/usr/bin/env python2
# -*- coding: utf-8 -*-
"""
Created on Mon Jun 12 09:56:30 2017
@author: plopezex
"""

import numpy as np
import os
import sys
import path_loader as ploader
from joint_distribution_2 import joint_distribution_2
from virtual_scan import v_scan
import interval_classifier as i_class
import h5py

def get_spectrum2(parametros): #entrada por grupo tamaño, df y numero
    rotacion

    path_samples=ploder.samples_path
    path_rotaciones=ploder.samples_path+'/rotaciones'
    path_output_espectros=os.path.join(path_samples,
        'espectros/filtered')
    rotations = 100

    def escanear(proj):

        vscan= v_scan(proj)
        raw_CLD=vscan.line_scan()
        return raw_CLD
    for parametro in parametros:
        prs=[]
        jd=joint_distribution_2()
        fd=parametro[0]
        rpm=parametro[1]
        num_floc=parametro[2]
        d_name_filtered='CLD_fd_'+fd+'_'+rpm+'floc_'+
            num_floc+'_FILTERED.csv'
        all_raw=[]
        jd=joint_distribution_2()
        try:
            for rotation in xrange(rotations):
```

```

path_mini_rot= os.path.join(path_rotaciones,
                             'rot_'+rpm+'_'+fd)
num_rot='{03d}'.format(rotation)
name= 'rotation_fd_'+fd+'_'+rpm+'floc_'+
      num_floc+'_rot_num_'+ num_rot+'_.hdf5'
f = h5py.File(os.path.join(path_mini_rot, name), 'r')
projection=f['rotation'][:]
f.close()
prs.append(projection)

except Exception as ex:
    print ex
    pass

for pr in prs:
    raw_CLD=escanear(pr)
    all_raw.append(raw_CLD)
dist=[i_class.assign_intervals(a) for a in all_raw]
d_filtered=[scipy.signal.medfilt(a,17) for a in dist]

for r in xrange(100):
    jd.merge_distribution(d_filtered[r])
print "escaneado_ "+str(parametro)
np.savetxt(os.path.join(path_output_espectros,
                        d_name_filtered), jd.joint_dist)

```

GENERATE SPECTRA

El método *generate_spectra* tiene como función la realización del barrido virtual de una población de flóculos y la generación de los correspondientes archivos con los histogramas mediante un esquema de computación en paralelo al objeto de optimizar el proceso de generación de espectros. El método se invoca desde la línea de comandos mediante órdenes del estándar MPI (*Message Passing Interface* o *Interfaz de Paso de Mensajes*). La función carga una serie de parámetros identificativos del archivo que contiene las proyecciones generadas de un flóculo (grupo de tamaños al que pertenece, su número dentro del grupo y su dimensión fractal), e invoca a la función *get_spectrum2*, que realiza el barrido virtual y el almacenamiento en el archivo correspondiente. El método *generate_spectra* permite al interfaz MPI distribuir el trabajo de escaneo entre los cuatro núcleos del procesador con que cuenta la CPU en la que se realizó el trabajo. Las distribuciones de tamaño generadas con este script se asociaron a la dimensión fractal consignada para la generación del flóculo correspondiente y las parejas histograma- D_f se utilizaron para entrenar y seleccionar un modelo de regresión por bosques aleatorios, preparando los datos y empleando un programa muy similar al que se incluye en este anexo con el título *RFR selector*. Una vez seleccionado el modelo de regresión óptimo, el sistema se verificó con datos de las distribuciones de tamaño obtenidas directamente por la sonda FBRM.

```

#!/usr/bin/env python2
# -*- coding: utf-8 -*-
"""

```

Created on Mon Jun 12 09:56:30 2017

```
@author: plopezex
"""

import numpy as np
import os
from mpi4py import MPI
from get_spectrum2 import get_spectrum2

def main():

    comm= MPI.COMM_WORLD
    rank=comm.Get_rank()
    size=comm.Get_size()
    name =MPI.Get_processor_name()
    MPI.Get_processor_name()

    path_MPI='/home/plopezex/Python_scripts/EOLO/ART_2/MPI'
    parametros=np.load(os.path.join(path_MPI,'parametros.npy'))
    #agrupa los archivos de proyecciones de cada floculo
    #en subgrupos para distribuir a los nucleos del procesador

    ni=1
    n0=parametros[0:ni]
    n1=parametros[ni:2*ni]
    n2=parametros[2*ni:3*ni]
    n3=parametros[3*ni:4*ni+1]

    if rank== 0: #asigna los trabajos a cada nucleo del
        #procesador
        get_spectrum2(n0)

    if rank== 1:
        get_spectrum2(n1)

    if rank== 2:
        get_spectrum2(n2)

    if rank== 3:
        get_spectrum2(n3)

    comm.Disconnect()
    # MPI.Finalize()
    os._exit(1)

if __name__ == '__main__':
    main()
```

ANEXO II

Publicaciones originales

PUBLICACIÓN I

Patricio Lopez-Exposito, Angeles Blanco, Carlos Negro (2016). Estimation of *Chlamydomonas reinhardtii* biomass concentration from chord length distribution data. *J. Appl Phyco*, 28(4), 2315-2322. (*JCR, Marine and Fresh Water Biology*, 24 de 106, Q1)

PUBLICACIÓN II

Patricio López Expósito, Angles Blanco, Carlos Negro. (2017). Laser reflectance measurement for the online monitoring of *Chlorella sorokiniana* biomass concentration. *Journal of Biotechnology*, 243, 10-15.(*JCR, Biotechnology & Applied Microbiology*, 65 de 160, Q2)

PUBLICACIÓN III

Lopez-Exposito, P., Campano, C., Blanco, A., & Negro, C. (2018). Microalgae harvesting by the novel flocculant hairy cationic nanocrystalline cellulose. *Colloids and Surfaces B. Biointerfaces*, 178, 329-336. (JCR, Biophysics, 13 de 72, Q1)

PUBLICACIÓN IV

López Expósito, P., Blanco, A., & Negro, C. (2017). Estimating fractal dimension of microalgal flocs through confocal laser scanning microscopy and computer modelling. *Algal Research*, 28, 74-79. (*JCR, Biotechnology & Applied Microbiology*, 36 de 160, Q1)

PUBLICACIÓN V

Lopez-Exposito, P., Blanco, A., & Negro, C. (2019). Direct estimation of microalgal flocs fractal dimension through laser reflectance and machine learning. *Algal Research*, 37, 240-247. (*JCR, Biotechnology & Applied Microbiology*, 36 de 160, Q1)

